NONLINEAR MULTISTEP METHODS FOR SOLVING INITIAL VALUE PROBLEMS IN ORDINARY DIFFERENTIAL EQUATIONS

Ding Lee

Naval Underwater Systems Center
New London, Connecticut

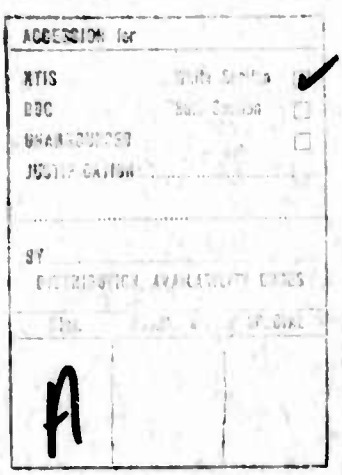
24 May 1974

PREFACE

This study was accepted in June 1974 as a dissertation in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Mathematics at the Polytechnic Institute of New York. The author wishes to thank his advisor, Professor Stanley Preiser, and the other members of the Guidance Committee.

REVIEWED AND APPROVED: 24 May 1974

R. M. Dunlap
Director, Plans and Analysis



The author of this report is located at the New London Laboratory, Naval Underwater Systems Center, New London, Connecticut 06320.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

AD-780 779

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR 4775	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* NONLINEAR MULTISTEP METHODS FOR SOLVING INITIAL VALUE PROBLEMS IN ORDINARY DIFFERENTIAL EQUATIONS		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Ding Lee		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Underwater Systems Center New London Laboratory New London, Connecticut 06320		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Underwater Systems Center Newport, Rhode Island 02840		12. REPORT DATE 24 May 1974 13. NUMBER OF PAGES 100
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Repo.)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Nonlinear multistep	Stiff systems	Consistency
Linear multistep	Stability	Convergence
Nonlinear operator	Strong stability	Discretization error
Stable matrix	Asymptotic stability	

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This thesis develops a family of Nonlinear Multistep (NLMS) numerical methods which solve initial value problems for systems of first-order differential equations. These methods are demonstrated to be a generalization of Linear Multistep (LMS) methods and are formulated to be particularly effective for equations whose solutions are asymptotically stable. The formal theory of NLMS methods with regard to stability,

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

20. (Cont'd)

consistency, and convergence is fully developed and proved. NLMS methods are strongly stable and accommodate A-stability in the sense of Dahlquist. Extensive numerical test results produced by NLMS methods show important advantages over Adams' and Gear's methods and Ehle's test results.

CONTENTS

CONTENTS (Cont'd)

LIST OF TABLES

LIST OF ILLUSTRATIONS

1. INTRODUCTION

Classical initial value problems of systems of first-order ordinary differential equations are designed to solve

$$\mathbf{y}' = \mathbf{f}(t,\mathbf{y}); \quad \mathbf{y}(0) = \mathbf{y}_0 . \tag{1.1}$$

Conventional methods such as Runge-Kutta and linear multistep (LMS) methods have been very well developed (Henrici [16]) and have been demonstrated effective for solving (1.1) when $\left\|\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right\|$ is small. Frequently, for $\left\|\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right\|$ large, prohibitively small step size values (h) are required for accuracy; then, conventional methods seem impractical. To overcome this difficulty, we choose to write (1.1) in the following way:

$$\mathbf{y}' = \mathbf{A}\mathbf{y} + \mathbf{g}(t,\mathbf{y}); \quad \mathbf{y}(0) = \mathbf{y}_0 , \tag{1.1}'$$

where $\mathbf{f}(t,\mathbf{y})$ may be written as $\mathbf{A}\mathbf{y} + \mathbf{g}(t,\mathbf{y})$ and $\mathbf{A}$ is either a constant matrix or a function of t. In the case where $\mathrm{Re}\{\lambda(\mathbf{A})\} < 0$ and $\lambda(\mathbf{A})$, the eigenvalues of $\mathbf{A}$, differ greatly in magnitude and $\mathbf{g}(t,\mathbf{y})$ is a slowly varying function in t, equation (1.1)' is called a "stiff" equation. These stiff equations frequently occur in the applications to chemical kinetics, reactor calculations, missile guidance, etc.

The search for effective schemes to solve stiff equations began over two decades ago and still goes on. Curtiss & Hirshfelder [9] encountered the stiff phenomenon in their study of chemical kinetics and proposed low-order multistep formulas to integrate scalar stiff equations. Cohen [8], in solving reactor kinetics equations, presented a generalization of Runge-Kutta methods. Certaine [7] demonstrated that if conventional schemes, such as trapezoidal rule, were used to solve (1.1)', then two problems were encountered — step size and accuracy. Certaine then proposed a method to handle scalar stiff equations that have short

time constants. Dahlquist [10] discussed a general treatment of the stability of linear multistep methods and investigated the special stability problem in connection with stiff equations. Dahlquist introduced an important concept, A-stability, and proved that there do not exist A-stable methods among linear multistep methods of order higher than 2. Widlund [32] and Gear [14] relaxed the A-stability concept in an attempt to create higher order multistep formulas suitable for stiff equations. Widlund [32] defined $A(\alpha)$-stability and showed that there exist K-step methods of order K which are $A(\alpha)$-stable for any $\alpha \leq \pi/2$ and $K \leq 4$. Gear [14] weakened the A-stability concept, defined stiff-stability, and derived stiffly stable methods of order ≤ 6. Gear [14] designed a computer program to perform such calculations. Stiffly stable methods of orders 7 and 8 have been found by Dill (1969) and of order up to 11 by Jain (1970), but no tests have been made on their algorithms. Lawson [24] generalized the Runge-Kutta method, Norsett [28] generalized the Adams-Bashforth methods, and Bjurel [3] modified linear multistep methods. Contributions also have been made by Miranker [27] and Guderley et al. [15], who designed stiff methods to solve special types of stiff equations.

The theory for stiff systems lacks a cohesiveness that this thesis attempts to achieve by providing:

(1) A complete formulation of nonlinear multistep (NLMS) methods, which is demonstrated to be a generalization of linear multistep methods both in technique and in theory.

(2) A full development and proof of the theory of NLMS methods with regard to stability, consistency, and convergence.

(3) A proof that NLMS methods accommodate A-stability in the sense of Dahlquist [10].

(4) A study of the effect of the error function C_{p+1} (as defined in section 4, Computational Considerations) by means of a perturbation of the characteristic roots. The study shows that LMS methods of order p that possess the smallest error C_{p+1} are only weakly stable. However, it will be indicated that there always exists a NLMS family possessing the smallest error C_{p+1}.

(5) Extensive tests of NLMS methods applied to a set of selected scalars and systems of stiff equations. Results are compared with Adams' methods [16], Gear's program [14], Seinfeld's paper [29] and Ehle's research report [11], and it is shown that NLMS has definite advantages over the above techniques.

(6) A section of conclusions and a summary of remaining problems with some suggested solutions.

(7) A listing of computer programs used to implement the NLMS methods.

2. PRELIMINARY CONSIDERATIONS

In this section, we define the problems under consideration and state the theory in relation to the existence and uniqueness of the solutions of approximating difference equations; the proofs for the existence and uniqueness theorem can be found in references [16] and [18]. The starting procedure involves the use of initial data; the solution of the difference equations depends continuously on the initial data. The order of the multistep methods will be defined as they are discussed and developed in section 3, Theory.

2.1. Problems Considered

In this paper, we consider the initial value problems of a system of first-order ordinary differential equations of the form:

$$\begin{aligned} \mathbf{y}' &= \mathbf{A}\mathbf{y} + \mathbf{g}(t,\mathbf{y}) \\ &= \mathbf{f}(t,\mathbf{y}) \qquad ; \quad \mathbf{y}(0) = \mathbf{y}_0 \end{aligned} \tag{2.1}$$

in the region R, defined by $0 = a \leq t \leq b < \infty$; $\|\mathbf{y}\| < \infty$. $\mathbf{A}$ is either a constant matrix or a function of t; consequently, a portion of the theory will be restricted to the important case, where the differential equations are stiff, i.e., $\mathrm{Re}\{\lambda(\mathbf{A})\} < 0$. Among the numerical test examples, $\mathbf{A}$ is chosen to be a constant matrix and $\mathrm{Re}\{\lambda(\mathbf{A})\} < 0$. The function $\mathbf{g}(t,\mathbf{y}) \in C^{p+1}$ $(p \geq 0)$ satisfies the Lipschitz condition,

$$\|\mathbf{g}(t,\mathbf{y}^*) - \mathbf{g}(t,\mathbf{y})\| \leq L\,\|\mathbf{y}^* - \mathbf{y}\| . \tag{2.2}$$

For the most interesting applications (those restricting the step size to conventional methods) $\rho(\mathbf{A}) \gg L$, where $\rho(\mathbf{A})$ is the spectral radius of $\mathbf{A}$.

2.2 Existence and Uniqueness Theorem

We assume that our initial value problems satisfy the conditions required by the existence and uniqueness theorem. We state the existence and uniqueness theorem expressed with respect to $\mathbf{f}(t,\mathbf{y})$ as follows:

Theorem 2.2: (Existence and Uniqueness Theorem)

We assume that $\mathbf{f}(t,\mathbf{y})$ satisfies the following two conditions:

(1) $\mathbf{f}(t,\mathbf{y})$ is continuous in R, where R is the region

$0 = a \leq t \leq b < \infty, \|\mathbf{y}\| < \infty.$

(2) $\exists$ a Lipschitz constant L^* $\ni$ for arbitrary $t \in [a,b]$ and any two vectors $\mathbf{y}$ and $\mathbf{y}^*$, the following condition is satisfied:

$$\|\mathbf{f}(t,\mathbf{y}^*) - \mathbf{f}(t,\mathbf{y})\| \leq L^* \|\mathbf{y}^* - \mathbf{y}\| . \tag{2.3}$$

Then, for any given initial vector $\mathbf{y}_0$, $\exists$ one and only one $\mathbf{y}(t)$ $\ni$

(1) $\mathbf{y}(t)$ is continuous and continuously differentiable for $t \in [a,b]$

(2) $\mathbf{y}'(t) = \mathbf{f}(t,\mathbf{y})$, $t \in [a,b]$

(3) $\mathbf{y}(a) = \mathbf{y}_0$.

To the differential equations (1.1), we adjoin appropriate relations, called initial conditions, that serve to define a "meaningful problem." If the solution of (1.1) satisfies appropriate initial conditions of smoothness, the problem (1.1) is termed well posed in the sense of Hadamard (Isaacson [20]), and the problem has a bounded, unique solution. Hochstadt [18] proved that the solution of the differential equation depends continuously on the initial data. This then fulfills the Hadamard well-posed statement. It ought to be pointed out that even though Hadamard's well-posed criterion is fulfilled, conventional techniques fail where

there is a very large Lipschitz constant. This is the area where we need to consider nonlinear multistep methods.

2.3. Norms

In a finite n-dimensional vector space, we define the p-norm of a vector to be

$$\|\mathbf{x}\|_p = \left(\sum_{j=1}^{n} |x_j|^p\right)^{1/p}$$

and

$$\|\mathbf{x}\|_1 = \sum_{j=1}^{n} |x_j|$$

$$\|\mathbf{x}\|_2 = \left(\sum_{j=1}^{n} |x_j|^2\right)^{1/2}$$

$$\|\mathbf{x}\|_\infty = \lim_{p\to\infty} \|\mathbf{x}\|_p = \max_j |x_j| .$$

For a matrix $\mathbf{A}$ of order n, we denote

$\lambda(\mathbf{A})$ as the eigenvalues of $\mathbf{A}$ and

$\rho(\mathbf{A})$ as the spectral radius of $\mathbf{A}$.

The different norms of $\mathbf{A}$ take the following definitions:

$$\|\mathbf{A}\|_s = [\rho(\mathbf{A}^*\mathbf{A})]^{1/2}$$

$$\|\mathbf{A}\|_1 = \max_{1\le j\le n} \sum_{i=1}^{n} |a_{ij}|$$

$$\|\mathbf{A}\|_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|$$

$$\|\mathbf{A}\|_E = \left(\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2\right)^{1/2} .$$

Let $\|\cdot\|_A$ indicate one norm and let $\|\cdot\|_B$ indicate another norm. Then $\|\cdot\|_A$ and $\|\cdot\|_B$ are said to be <u>equivalent</u> if $\exists$ two positive numbers a and b $\ni$

$$a\|\cdot\|_A \leq \|\cdot\|_B \leq b\|\cdot\|_A .$$

We know that in a finite-dimensional space, all norms are equivalent; therefore, the norms used in this paper do not refer to any specific norm. However, in the Pade approximation, we use the column norm $\|\cdot\|_1$; in the Ehle's test examples, we use $\|\cdot\|_2$; and in the other test examples, we use $\|\cdot\|_\infty$. In the lemmas and theorems where the norm does not have a subscript we mean any norm.

3. THEORY

This section develops the full theory that provides the basis for NLMS methods, including the development of the important notions of Stability and Consistency. The formulation of NLMS methods, which assures consistency and is closely connected with the theory, is also described in this section.

NLMS methods are demonstrated to be a generalization of LMS methods. The general formula is described by (3.5), and the formulation is expressed by (3.36). Both explicit and implicit schemes are given in matrix expressions up to an order of 3. The application of NLMS methods for the solution of stiff ordinary differential equations leads naturally to the selection of strongly stable methods.

The theorem on convergence (Theorem 3.7), which follows for such strongly stable and consistent NLMS integrators, is also presented in this section. Three lemmas needed to prove the convergence theorem are developed. Some of these proofs are a modification and extension of the proofs used by Henrici [17] for LMS methods. The theorem of A-stability is also presented in this section.

3.1. Nonlinear Multistep (NLMS) Algorithm

For convenience, assume $\mathbf{A}$ to be a nonsingular, constant matrix. Equation (1.1)' can be written as

$$\frac{d}{dt}\left(e^{-\mathbf{A}t}\mathbf{y}\right) = e^{-\mathbf{A}t}\mathbf{g}(t,\mathbf{y}); \quad \mathbf{y}(0) = \mathbf{y}_0. \tag{3.1}$$

Then, integration of the above equation over the interval $[t_n, t_{n+i}]$ gives

$$\mathbf{y}(t_{n+i}) = e^{i\mathbf{A}h}\,\mathbf{y}(t_n) + \int_{t_n}^{t_{n+i}} e^{\mathbf{A}(t_{n+i}-t')}\,\mathbf{g}(t',\mathbf{y})\,dt'. \tag{3.2}$$

If $e^{-\mathbf{A}t}\mathbf{g}(t,\mathbf{y})$ is a slowly varying function, a simple change of variable allows successful integration by conventional methods. For $\mathbf{g}(t,\mathbf{y})$ slowly varying and $\mathrm{Re}\{\lambda(\mathbf{A})\} < 0$, $\rho(\mathbf{A}) \gg 1$, conventional methods require a prohibitively small step size,which we overcome by NLMS methods. Our method of attack is to express $\mathbf{g}(t,\mathbf{y})$ as a low-order polynomial in t by retaining the first few terms of the Taylor series of $\mathbf{g}(t,\mathbf{y})$ expanded about t_n. The complete derivation of this idea is properly described in section 3.3, Consistency. When $\mathbf{g}(t,\mathbf{y}) = \mathbf{0}$, equation (1.1)' becomes homogeneous, i.e.,

$$\mathbf{y}' = \mathbf{A}\mathbf{y}; \quad \mathbf{y}(0) = \mathbf{y}_0. \tag{3.3}$$

Without loss of generality, we consider $a = t_0 = 0$. The solution of (3.3) is $\mathbf{y}(t) = e^{\mathbf{A}t}\mathbf{y}(0)$. Consequently,

$$\mathbf{y}(t_{n+i}) = e^{i\mathbf{A}h}\mathbf{y}(t_n),$$

which is the rigorous solution in the absence of round-off errors, where $i = 0, 1, 2, \ldots$, an integer index, and $h = t_{n+1} - t_n$.

The linear multi-K-step methods take the general form

$$\sum_{i=0}^{K} \alpha_i \mathbf{y}_{n+i} = h \sum_{i=0}^{K} \beta_i \mathbf{f}_{n+i}, \tag{3.4}$$

where $\alpha_K \neq 0$ and $|\alpha_0| + |\beta_0| > 0$.

If $\begin{cases} \beta_K = 0 \\ \beta_K \neq 0 \end{cases}$, the method is $\begin{cases} \text{explicit} \\ \text{implicit} \end{cases}$.

The generalization of (3.4) leads to

$$\sum_{i=0}^{K} \alpha_i e^{\mathbf{A}h(K-i)} \mathbf{y}_{n+i} = h \sum_{i=0}^{K} \phi_{Ki}(\mathbf{A}h)\, \mathbf{g}_{n+i}, \tag{3.5}$$

where $\alpha_K \neq 0$ and $|\alpha_0| + |\lambda(\phi_{K0}(\mathbf{A}h))| > 0$.

If $\left\{\begin{array}{l}\phi_{KK}(\mathbf{A}h) = \mathbf{0} \\ \phi_{KK}(\mathbf{A}h) \neq \mathbf{0}\end{array}\right.$, the method is $\left\{\begin{array}{l}\text{explicit} \\ \text{implicit}\end{array}\right.$.

Without loss of generality, we assume $\alpha_K = 1$ for computational convenience. The coefficients of $\mathbf{y}_{n+i}$, $\mathbf{g}_{n+i}$ depend upon $(\mathbf{A}h)$. We note that we need K starting values to proceed.

3.1.1. Starting Procedure

A convergent K-step method will produce a uniquely determined sequence $\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_n$ for an arbitrary set of starting vectors $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{K-1}$. In practice, we obtain the starting procedure by setting the starting vectors equal to the given initial vector and calculating the subsequent (K-1) vectors. These starting vectors are required to be bounded in order to meet the stability criterion. We have already shown that if the differential equation satisfies certain requirements, the solutions of the difference equation depend continuously on the initial data. Thus, a unique solution exists and the numerical solution approaches the exact solution.

3.2. Stability

If a bounded starting procedure yields a uniformly bounded solution of the approximating difference equation to the differential equation (1.1) as $h \to 0$, we say that the method is <u>stable</u>. We can also describe a stable method as follows: Let $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{K-1}$ denote the K initial vectors ,

$$\|\mathbf{s}_i(h)\| < M, \text{ a constant.}$$

Let

$$\mathbf{y}_i = \mathbf{S}_i(h) \; ; \quad i = 0, 1, 2, \ldots , K-1 \; .$$

Then $\exists$ a constant M', independent of h, $\ni$

$$\max_n \| \mathbf{y}_n(h) \| < M' .$$

The stability is determined by the root condition, i.e., the modulus of the roots of every characteristic polynomial must not exceed 1 and the roots of modulus 1 must be simple. (The characteristic polynomial is discussed in this section.) We now proceed to develop the theory with regard to the concept of stability.

If equation (1.1)' is homogeneous, i.e., $\mathbf{g}(t, \mathbf{y}) = \mathbf{0}$, we expect that the values $\mathbf{y}(t_n)$ can be found exactly in the absence of round-off error. If (3.5) is to hold when

$$\mathbf{y}(t_{n+i}) = e^{iAh} \mathbf{y}(t_n) \; , \tag{3.6}$$

then substituting (3.6) into (3.5) gives

$$\sum_{i=0}^{K} \alpha_i \, e^{Ah(K-i)} \; e^{iAh} \mathbf{y}(t_n) \; = e^{AhK} \mathbf{y}(t_n) \sum_{i=0}^{K} \alpha_i . \tag{3.7}$$

For $\mathbf{y}(t_n) \neq \mathbf{0}$ and (3.7) to be zero, we discover that $\sum_{i=0}^{K} \alpha_i$ must be 0. This is identical to the necessary condition for LMS methods to be consistent. The characteristic polynomial, following Henrici [16] and Dahlquist [10], for LMS methods is expressed by

$$\rho(\zeta) = \sum_{i=0}^{K} \alpha_i \zeta^i . \tag{3.8}$$

Polynomial $\rho(\zeta)$ is said to satisfy the "root condition" if K roots ζ_i satisfy $|\zeta_i| \leq 1$ and if the roots satisfying $|\zeta_i| = 1$ have multiplicity 1. Later in this section, we show that the characteristic polynomial of NLMS methods generalizes

the characteristic polynomial of LMS methods and that our particular choice of NLMS methods obeys the root condition. We will explore the necessary condition of stability later in this section and the sufficient condition of stability in section 3.6.3. Since 1 is a simple root of the characteristic polynomial, we have

$$\rho(1) = \sum_{i=0}^{K} \alpha_i = 0, \tag{3.9}$$

which is imposed as one of the conditions of consistency for LMS methods.

Formula (3.7) can be written as

$$\sum_{i=0}^{K} \alpha_i e^{\mathbf{A}h(K-i)} (e^{\mathbf{A}h}\zeta)^i = e^{\mathbf{A}hK} \sum_{i=0}^{K} \alpha_i \zeta^i = e^{\mathbf{A}hK} \rho(\zeta).$$

If all K roots satisfy the root condition, $\rho(\zeta)$ must be equal to 0. Then

$$\sum_{i=0}^{K} \alpha_i e^{\mathbf{A}h(K-i)} (e^{\mathbf{A}h}\zeta)^i = \mathbf{0}.$$

Since a matrix annihilates its characteristic polynomial, its eigenvalues must also annihilate the same polynomial. Thus the above formula can be written as

$$\rho(\lambda, \zeta) = \sum_{i=0}^{K} \alpha_i^{\lambda(\mathbf{A})h(K-i)} (e^{\lambda(\mathbf{A})h}\zeta)^i = \sum_{i=0}^{K} \alpha_i (e^{\lambda hK}\zeta^i) = \mathbf{0}. \tag{3.10}$$

Equation (3.10) is a set of n equations for the components of $e^{\lambda hK}\zeta$. Each equation is a characteristic polynomial of the form $\sum_{i=0}^{K} \alpha_i \xi^i = 0$, where ξ, which stands for $\xi_j = e^{\lambda_j hK}\zeta_j$, is the component of the j-th characteristic polynomial and $\xi_{j,1}, \xi_{j,2}, \ldots, \xi_{j,K}$ are K roots of the j-th characteristic polynomial $\rho(\lambda, \zeta)$ with $\lambda = \lambda_j(\mathbf{A})$. Thus, we have

$$\rho(\lambda, 1) = \sum_{i=0}^{K} \alpha_i (e^{\lambda hK}) = e^{\lambda hK} \sum_{i=0}^{K} \alpha_i = e^{\lambda hK} \rho(1) = 0. \tag{3.11}$$

This is a condition of consistency for LMS methods. Note that (3.10) generalizes (3.8) and (3.11) generalizes (3.9). Equation (3.10) is defined as the characteristic polynomial for NLMS methods, and (3.11) is a necessary condition of consistency for NLMS methods.

Let us use test problem 3 (from section 5, Numerical Comparisons) as an example to show that the root condition is a necessary condition for stability. Consider the problem

$$\mathbf{y}' = \begin{pmatrix} -1 & 95 \\ -1 & -97 \end{pmatrix}\mathbf{y}; \quad \mathbf{y}^{(0)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Using the nonlinear multi-2-step method with $\alpha_0 = 4$, $\alpha_1 = -5$, and $\alpha_2 = 1$, we obtain

$$4e^{2\mathbf{A}h}\mathbf{y}_n - 5e^{\mathbf{A}h}\mathbf{y}_{n+1} + \mathbf{y}_{n+2} = \mathbf{0},$$

with the initial values

$$\mathbf{y}_0 = (1,\ 1)^T$$

$$\mathbf{y}_1 = (.5791054,\ -.60958467E\text{-}02)^T$$

and the step size, h, = .625. Our NLMS characteristic polynomial is

$$\rho(\lambda, \zeta) = 4e^{2\lambda h} - 5e^{2\lambda h}\zeta + e^{2\lambda h}\zeta^2$$

$$= \mathbf{0} = e^{2\lambda h}(\zeta - 1)(\zeta - 4),$$

which has two simple roots, 1 and 4. Obviously, this violates the root condition. As we proceed to solve this problem numerically, we can see, from the computer results shown below, that the divergence becomes evident after 7 steps. The first column of the results is the iteration index, and the second column is the current t values. The last two columns show the calculated numerical values of two arguments, indicated by A. The two values below the arguments are exact solution values, indicated by T.

1	.12500000+01	.16589938+00	-.17463091-02 A
		.16591649+00	-.17464893-02 T
2	.18750000+01	.47506461-01	-.50006794-03 A
		.47535869-01	-.50037757-03 T
3	.25000000+01	.13581334-01	-.14296139-03 A
		.13619255-01	-.14336057-03 T
4	.31250000+01	.38569114-02	-.40599063-04 A
		.39019817-02	-.41073492-04 T
5	.37500000+01	.10657062-02	-.11217958-04 A
		.11179365-02	-.11767753-04 T
6	.43750000+01	.26023676-03	-.27393339-05 A
		.32029417-03	-.33715176-05 T
7	.50000000+01	.22870184-04	-.24073868-06 A
		.91765814-04	-.96595595-06 T
8	.56250000+01	-.52688805-04	.55461894-06 A
		.26291346-04	-.27675101-06 T
9	.62500000+01	-.82990288-04	.87358188-06 A
		.75325968-05	-.79290492-07 T
10	.68750000+01	-.10158864-03	.10693539-05 A
		.21581251-05	-.22717106-07 T
11	.75000000+01	-.11828226-03	.12450763-05 A
		.61831319-06	-.65085600-08 T
12	.81250000+01	-.13609010-03	.14325272-05 A
		.17714970-06	-.18647337-08 T
13	.87500000+01	-.15611957-03	.16433637-05 A
		.50754238-07	-.53425514-09 T
14	.93750000+01	-.17896583-03	.18838506-05 A
		.14541333-07	-.15306666-09 T
15	.10000000+02	-.20511784-03	.21591349-05 A
		.41661615-08	-.43854332-10 T
16	.10625000+02	-.23508066-03	.24745329-05 A
		.11936253-08	-.12564477-10 T
17	.11250000+02	-.26941724-03	.28359707-05 A
		.34197936-09	-.35997828-11 T

The nonlinear multi-2-step method used to solve problem 3 is stable because we chose $\alpha_0 = 0$, $\alpha_1 = -1$, and $\alpha_2 = 1$, so that the characteristic polynomial has two distinct roots, 0 and 1. Hence, the root condition is satisfied, and the method is stable. Thus, we have shown that the root condition is a necessary condition for stability. We defer the proof of the sufficient condition to lemma 3.6.3. Thus, the stability for NLMS methods is a direct generalization of the stability of LMS methods (Henrici [16]).

3.2.1. Strong Stability

A particularly advantageous feature of NLMS methods is that the strong stability condition results when $\mathrm{Re}\{\lambda(\mathbf{A})\} < 0$. Since stiff differential equations frequently occur when $\mathrm{Re}\{\lambda(\mathbf{A})\} < 0$ with $\rho(\mathbf{A}) >> 1$, it is most important that the parasitic growth of the extraneous solution of the difference equations be damped out. To ensure this, the NLMS methods are selected to be strongly stable. This is not a restriction of NLMS methods since the methods are also applicable when $\mathrm{Re}\{\lambda(\mathbf{A})\} \geq 0$; in this case, the error growth is appraised by the estimate furnished by lemma 3.6.3.

A measure of the growth of LMS methods is provided by examining the solution of the homogeneous, constant coefficient difference equation. Consider the homogeneous equation of LMS methods,

$$\sum_{i=0}^{K} \alpha_i \mathbf{y}_{n+i} = \mathbf{0}. \tag{3.12}$$

We recall that strongly stable solutions of the LMS methods occur when

$$\rho(\zeta) = \sum_{i=0}^{K} \alpha_i \zeta^i ,$$

$$\rho_1(\zeta) = \frac{\rho(\zeta)}{\zeta - 1} \text{ has roots } \zeta_2, \zeta_3, \ldots, \zeta_K,$$

where

$$|\zeta_2|, |\zeta_3|, \ldots, |\zeta_K| < 1.$$

The corresponding homogeneous equation of NLMS methods is

$$\sum_{i=0}^{K} \alpha_i e^{\mathbf{A}h(K-i)} \mathbf{y}_{n+i} = \mathbf{0}. \tag{3.13}$$

Since

$$e^{\mathbf{A}h(K-i)} \neq \mathbf{0},$$

let

$$e^{\mathbf{A}h(K-i)} \mathbf{y}_{n+i} = \mathbf{w}_{n+i}.$$

Then (3.13) becomes

$$\sum_{i=0}^{K} \alpha_i \mathbf{w}_{n+i} = \mathbf{0}. \tag{3.14}$$

Equation (3.14), of course, has precisely the same constant coefficients as the linear difference equations of order K. In fact,

$$\begin{aligned} \mathbf{w}_{n+i} &= e^{\mathbf{A}h(K-i)} \mathbf{y}_{n+i} \\ &= e^{\mathbf{A}h(K-i)} (e^{\mathbf{A}h} \zeta_j)^i \\ &= e^{\mathbf{A}hK} \zeta_j^i; \quad j = 1, 2, \ldots, K. \end{aligned}$$

Strongly stable solutions will result for

$$\left| e^{\lambda(\mathbf{A})hK} \zeta_j \right| < 1.$$

Note that these extraneous solutions damp out extremely fast since $\mathrm{Re}\{\lambda(\mathbf{A})\} < 0$. Therefore, we see that strong stability implies that $\| e^{\mathbf{A}h(K-i)} \mathbf{y}_{n+i} \|$ is uniformly

bounded. For $\mathbf{A} = \mathbf{0}$, $\left\| e^{\mathbf{A}h(K-i)} \mathbf{y}_{n+i} \right\|$ reduces to $\|\mathbf{y}_{n+i}\|$, which means that $\|\mathbf{y}_{n+i}\|$ is uniformly bounded; this is the stability definition for LMS methods.

3.2.2. A-Stability

Another advantageous feature of NLMS methods is that when they are used to solve stiff equations, they are A-stable. Dahlquist [10] defined a method to be A-stable if the numerical solution $\|\mathbf{y}_n\| \to 0$ asymptotically as $n \to \infty$ for the differential equation $\mathbf{y}' = \mathbf{A}\mathbf{y}$ where $\mathrm{Re}\{\lambda(\mathbf{A})\} < 0$. If A-stable methods of order higher than 2 exist, they do not belong to the linear multistep family since it has been proved by Dahlquist [10] that an LMS method of order higher than 2 cannot be A-stable. However, since A-stability is a desirable property when solving stiff equations, it is preferable that it be retained in NLMS methods. We will introduce a theorem which shows that NLMS methods accommodate the A-stability in the sense of Dahlquist.

Matrix Exponential

We begin by discussing the computation of a matrix exponential, $e^{\mathbf{A}}$. If $\mathbf{A}$ is a scalar, there is no difficulty in computing $e^{\mathbf{A}}$. If $\mathbf{A}$ is a stable matrix (Young [33]), i.e., $\mathrm{Re}\{\lambda(\mathbf{A})\} < 0$, then the rational Pade approximation is also stable (Varga [31] and Lawson [24]), as shown by the following lemma.

Lemma 3.2.2. (Pade Lemma)

Denote the Pade approximation to $e^{\mathbf{A}h}$ by Pade $(\mathbf{A}h)$. Then for $\mathrm{Re}\{\lambda(\mathbf{A})\} < 0$, the Pade $(\mathbf{A}h)$ is stable, i.e., $\rho(\text{Pade }(\mathbf{A}h)) < 1$.

The Pade approximation to a matrix exponential, $e^{\mathbf{A}}$, has many different expressions, which can be found in references [5] and [31]. In our present test computations following Blue [5], we use

$$R_{2,2}(\mathbf{A}) = e^{\mathbf{A}} = \left[\mathbf{I} - \frac{\mathbf{A}}{2} + \frac{\mathbf{A}^2}{12}\right]^{-1}\left[\mathbf{I} + \frac{\mathbf{A}}{2} + \frac{\mathbf{A}^2}{12}\right]$$

for $\mathrm{Re}\{\lambda(\mathbf{A})\} < 0$. Where x is a real scalar <0, e^x can be calculated directly from an accurate exponential routine. The Pade approximation is applied with the requirement that $\rho(\mathbf{A}h) < 1$ when $\mathrm{Re}\{\lambda(\mathbf{A})\} < 0$. The Pade approximation to $e^{\mathbf{A}h}$ using a polynomial of degree n in the numerator and m in the denominator has an error $0(h^{n+m+1})$ as $h \to 0$ (Varga [31]). If $\rho(\mathbf{A}h)$ is not <1, the accuracy can be ensured by the identity $e^{\mathbf{A}h} = (e^{2^{-m}\mathbf{A}h})^{2^m}$.

Theorem 3.2.2. (A-Stability Theorem)

When used to solve stiff equations, NLMS methods accommodate the A-stability in the sense of Dahlquist.

Proof: In the Dahlquist sense, when applying NLMS methods to the problem $\mathbf{y}' = \mathbf{A}\mathbf{y}$, which implies $\mathbf{g}(t,\mathbf{y}) = \mathbf{0}$, NLMS methods produce the approximate solution to the problem: $\mathbf{y}_n = e^{\mathbf{A}t_n}\mathbf{y}_0 = e^{n\mathbf{A}h}\mathbf{y}_0$. Since the Pade $(n\mathbf{A}h)$ is stable, the $\lim_{n\to\infty} \|\mathbf{y}_n\| \to 0$, thus establishing the A-stability.

We note in passing that the solution to this problem is the principal reason for NLMS methods to be of interest. The NLMS methods solve this problem rigorously for every constant matrix $\mathbf{A}$ in the absence of round-off error.

3.3 Consistency

This section deals with the development of the theory of consistency in relation to NLMS methods. Later in this section, we show that NLMS methods are consistent and demonstrate that our consistency is a generalization of the consistency of LMS methods. An immediate need is to define what we mean by consistency. A method of (3.5) is said to be consistent if

$$\max_n \left\| \sum_{i=0}^{K} \alpha_i e^{\mathbf{A}h(K-i)} \mathbf{y}_{n+i} - h \sum_{i=0}^{K} \phi_{Ki}(\mathbf{A}h)\, \mathbf{g}_{n+i} \right\|$$

is small as $h \to 0$. We shall show that the consistency we will develop for NLMS methods actually satisfies our definition.

Problem (1.1)' can be written as (3.1), which is

$$\frac{d}{dt}(e^{-\mathbf{A}t}\mathbf{y}) = e^{-\mathbf{A}t}\mathbf{g}(t,\mathbf{y}); \quad \mathbf{y}(0) = \mathbf{y}_0.$$

Integration of the above system over the interval $[t_n, t_{n+i}]$ gives

$$\mathbf{y}(t_{n+i}) = e^{i\mathbf{A}h}\mathbf{y}(t_n) + \int_{t_n}^{t_{n+i}} e^{\mathbf{A}(t_{n+i}-t')}\mathbf{g}(t',\mathbf{y})\, dt',$$

which is our equation (3.2). If $e^{-\mathbf{A}t}\mathbf{g}(t,\mathbf{y})$ is a slowly varying function, a simple change of variable, i.e.,

$$\mathbf{z} = e^{-\mathbf{A}t}\mathbf{y},$$

allows successful integration by conventional methods such as the Runge-Kutta or LMS methods. Lawson [24] used this idea. For $\mathbf{g}(t,\mathbf{y})$ slowly varying,

Re $\{\lambda(\mathbf{A})\} < 0$, and $\rho(\mathbf{A}) >> 1$, conventional methods require a prohibitively small step size, which we overcome by NLMS methods. The method of attack is to express $\mathbf{g}(t,\mathbf{y})$ as a low-order polynomial in t, e.g., by retaining the first few terms of the Taylor series of $\mathbf{g}(t,\mathbf{y})$ expanded about t_n. For the moment, let us introduce the NLMS operator, $\mathcal{L}_N[\mathbf{y}(t);h]$. (The construction of such an operator is described in section 3.4, Nonlinear Operator.)

Write

$$\mathcal{L}_N[\mathbf{y}(t);h] = \sum_{i=0}^{K} \alpha_i e^{\mathbf{A}h(K-i)} \mathbf{y}(t+ih) - h \sum_{i=0}^{K} \phi_{Ki}(\mathbf{A}h)\mathbf{g}(t+ih,\mathbf{y}), \qquad (3.15)$$

and introduce the true operator,

$$\mathcal{L}[\mathbf{y}(t)] = \frac{d\mathbf{y}}{dt} - \mathbf{A}\mathbf{y} - \mathbf{g}(t,\mathbf{y}) = \mathbf{0}. \qquad (3.15)'$$

For $\mathbf{g}(t,\mathbf{y}) \in C^{p+1}$, we evaluate the local discretization error as follows:

$$\tau[\mathbf{y}(t);h] = \mathcal{L}_N[\mathbf{y}(t);h] - \mathcal{L}[\mathbf{y}(t)]$$

$$= \sum_{i=0}^{K} \alpha_i e^{\mathbf{A}h(K-i)} \mathbf{y}(t+ih) - h \sum_{i=0}^{K} \phi_{Ki}(\mathbf{A}h)\mathbf{g}(t+ih,\mathbf{y})$$

$$- \left\{\frac{d\mathbf{y}}{dt} - \mathbf{A}\mathbf{y} - \mathbf{g}(t,\mathbf{y})\right\}. \qquad (3.16)$$

The terms inside { } of (3.16) vanish because of (3.15)'. If we expand $\mathbf{g}(t,\mathbf{y})$ in a Taylor series expansion around t_n, we get

$$\mathbf{g}(t,\mathbf{y}) = \sum_{j=0}^{\infty} \frac{\mathbf{g}^{(j)}(t_n, \mathbf{y}(t_n))}{j!}(t-t_n)^j$$

$$= \sum_{j=0}^{p} \frac{\mathbf{g}^{(j)}(t_n, \mathbf{y}(t_n))}{j!}(t-t_n)^j + \frac{\mathbf{g}^{(p+1)}(t_n, \mathbf{y}(t_n))}{(p+1)!}(t-t_n)^{p+1}$$

$$+ \theta \mathbf{G}_{p+1}(t-t_n)\frac{(t-t_n)^{p+1}}{(p+1)!}. \qquad (3.17)$$

In the above,

$$G_{p+1}(t - t_n) = \max_{t-t_n \in Kh} \| g^{(p+1)}(t, y(t)) - g^{(p+1)}(t_n, y(t_n)) \|$$

is the modulus of continuity when $g(t,y)$ satisfies the Lipschitz condition only, $0 < \theta < 1$, and $p + 1 = 0$.

By substituting (3.17) into (3.2), we get

$$y(t_{n+i}) = e^{iAh} y(t_n) + \sum_{j=0}^{p} \int_{t_n}^{t_{n+i}} e^{-A(t'-t_{n+i})} \frac{g^{(j)}(t_n, y(t_n))}{j!} (t' - t_n)^j \, dt'$$

$$+ \int_{t_n}^{t_{n+i}} e^{-A(t'-t_{n+i})} \frac{g^{(p+1)}(t_n, y(t_n))}{(p+1)!} (t' - t_n)^{p+1} \, dt'$$

$$+ \theta \int_{t_n}^{t_{n+i}} e^{-A(t'-t_{n+i})} G_{p+1}(t' - t_n) \frac{(t' - t_n)^{p+1}}{(p+1)!} \, dt' . \qquad (3.18)$$

Define

$$I_i^j(Ah) = \int_{t_n}^{t_{n+i}} e^{-A(t'-t_{n+i})} (t' - t_n)^j \, dt' . \qquad (3.19)$$

Expanding $g(t+ih,y)$ at $t=t_n$ and neglecting the terms modulo $(p+1)$, we get

$$g(t + ih, y) = \sum_{j=0}^{p} \frac{(ih)^j}{j!} g^{(j)}(t_n, y) . \qquad (3.20)$$

By substituting (3.18) and (3.20) into (3.16) and using definition (3.19), we get

$$\tau[y(t); h] = \sum_{i=0}^{K} \alpha_i e^{Ah(K-i)} e^{iAh} y(t_n) + \sum_{j=0}^{p} \left\{ \sum_{i=0}^{K} \alpha_i e^{Ah(K-i)} \left[\frac{I_i^j(Ah)}{j!} \right] \right.$$

$$\left. - h \sum_{i=0}^{K} \frac{(ih)^j}{j!} \phi_{Ki}(Ah) \right\} g^{(j)}(t_n, y) + \sum_{i=0}^{K} \alpha_i e^{Ah(K-i)} \left[\frac{I_i^{p+1}(Ah)}{(p+1)!} \right.$$

$$\left. \cdot (g^{(p+1)}(t_n, y(t_n)) + \theta G_{p+1}(t - t_n)) \right] . \qquad (3.21)$$

Now, let us look at the last term of (3.21). For $\mathbf{g}(t, \mathbf{y}(t)) \in \mathcal{P}_p$, the class of polynomials of order p, the last term vanishes. For $\mathbf{g}(t, \mathbf{y}(t)) \in C^{p+1}$, the last term will be $O(h^{p+2})$; this is followed by examining the bound for the last term. The bound is given by

$$\left\| \sum_{i=0}^{K} \alpha_i \, e^{\mathbf{A}h(K-i)} \left[\frac{\mathbf{I}_i^{p+1}(\mathbf{A}h)}{(p+1)!} \right] \left(\mathbf{g}^{(p+1)}(t_n, \mathbf{y}(t_n)) + \theta \mathbf{G}_{p+1}(t - t_n) \right) \right\| \leq \sum_{i=0}^{K} |\alpha_i|$$

$$\cdot \left\| e^{\mathbf{A}h(K-i)} \right\| \frac{K^{p+2}}{(p+2)!} \left(\| \mathbf{g}^{(p+1)}(t_n, \mathbf{y}(t_n)) \| + \| \mathbf{G}_{p+1}(Kh) \| \right) h^{p+2} = O(h^{p+2}) \, . \quad (3.22)$$

As $h \to 0$, this bound vanishes.

For arbitrary $\mathbf{g}(t_n, \mathbf{y})$, we select $\{\ \}$ of (3.21) to be zero, so that $\lim_{h \to 0} \tau[\mathbf{y}(t); h] = \mathbf{0}$. This defines consistency for NLMS operators and shows that the NLMS operator is consistent with the true operator in the sense of Keller [22]. Since the true operator is $\mathbf{0}$, then the $\lim_{h \to 0} \tau[\mathbf{y}(t); h] = \mathbf{0}$ is equivalent to

$$\lim_{h \to 0} \max_{n} \left\| \sum_{i=0}^{K} \alpha_i \, e^{\mathbf{A}h(K-i)} \, \mathbf{y}_{n+i} - h \sum_{i=0}^{K} \phi_{Ki}(\mathbf{A}h) \, \mathbf{g}_{n+i} \right\| = 0 \, ,$$

which is our definition of consistency. The selection of $\{\ \}$ of (3.21) to be $\mathbf{0}$ enables us to determine $\phi_{Ki}(\mathbf{A}h)$ and guarantees the consistency, so that a formal analogy with LMS methods results. Thus, we see that our NLMS methods are a generalization of LMS methods.

3.4 Nonlinear Operator, $\mathcal{L}_N[\mathbf{y}(t); h]$

Since we have now established the consistency, let us define

$$\mathcal{J}_i^j = \frac{\mathbf{A}^{j+1}\, \mathbf{I}_i^j(\mathbf{A}h)}{j!} = \frac{\mathbf{A}^{j+1}}{j!} \int_{t_n}^{t_{n+i}} e^{-\mathbf{A}(t'-t_{n+i})} (t'-t_n)^j \, dt' .$$

For $j = 0, 1$, we get

$$\mathcal{J}_i^0 = \mathbf{A} \int_{t_n}^{t_{n+i}} e^{-\mathbf{A}(t'-t_{n+i})} \, dt' = e^{i\mathbf{A}h} - I$$

$$\mathcal{J}_i^1 = \mathbf{A}^2 \int_{t_n}^{t_{n+i}} e^{-\mathbf{A}(t'-t_{n+i})} (t'-t_n) \, dt' = (e^{i\mathbf{A}h} - I) - i\mathbf{A}h .$$

By induction, we get

$$\mathcal{J}_i^{m+1} = \mathcal{J}_i^m - \frac{(i\mathbf{A}h)^{m+1}}{(m+1)!} ;$$

then

$$\mathcal{J}_i^j = e^{i\mathbf{A}h} - \sum_{\ell=0}^{j} \frac{(i\mathbf{A}h)^\ell}{\ell!} . \tag{3.23}$$

We already have associated the nonlinear operator with the nonlinear multistep formula by (3.15). Take $n = 0$; then

$$\mathcal{L}_N[\mathbf{y}(t); h] = \sum_{i=0}^{K} \alpha_i \, e^{\mathbf{A}h(K-i)} \, \mathbf{y}_i - h \sum_{i=0}^{K} \phi_{Ki}(\mathbf{A}h) \, \mathbf{g}_i . \tag{3.24}$$

If we use formula (3.18) for $\mathbf{y}_i$ and formula (3.20) for $\mathbf{g}_i$ (letting $p \to \infty$) and formula (3.23), and substitute them into (3.24), we get

$$\mathcal{L}_N[\mathbf{y}(t); h] = \sum_{i=0}^{K} \alpha_i \, e^{\mathbf{A}h(K-i)} \left[e^{i\mathbf{A}h} \mathbf{y} + \sum_{j=0}^{\infty} \frac{\mathbf{I}_i^j(\mathbf{A}h)}{j!} \mathbf{g}^{(j)} \right] - h \sum_{i=0}^{K} \phi_{Ki}(\mathbf{A}h) \left[\sum_{j=0}^{\infty} \frac{(ih)^j}{j!} \mathbf{g}^{(j)} \right]$$

$$= \left\{ \sum_{i=0}^{K} \alpha_i \, e^{\mathbf{A}h(K-i)} \, e^{i\mathbf{A}h} \mathbf{y} \right\} + \mathbf{C}_0(h)\mathbf{g} + \mathbf{C}_1(h)\mathbf{g}' + \ldots + \mathbf{C}_q(h)\mathbf{g}^{(q)} + \ldots . \tag{3.25}$$

Note that because of the root condition, $\{\quad\}$ of (3.25) $\to \mathbf{0}$. Thus,

$$\mathcal{L}_N[\mathbf{y}(t);h] = \sum_{m=0}^{\infty} \mathbf{C}_m \mathbf{g}^{(m)}, \tag{3.26}$$

where

$$\mathbf{C}_j(h) = \sum_{i=0}^{K} \alpha_i e^{\mathbf{A}h(K-i)} \left[\frac{\mathbf{I}_i^j(\mathbf{A}h)}{j!}\right] - h \sum_{i=0}^{K} \frac{(ih)^j}{j!} \phi_{Ki}(\mathbf{A}h), \quad j = 0, 1, \ldots. \tag{3.27}$$

We define the order p of NLMS methods to mean that

$$\mathbf{C}_0(h) = \mathbf{C}_1(h) = \ldots = \mathbf{C}_p(h) = \mathbf{0} \text{ but } \mathbf{C}_{p+1} \neq \mathbf{0}.$$

At this point, let us distinguish the orders between LMS and NLMS, i.e., p_{LMS} and p_{NLMS}. When $\mathbf{A} = \mathbf{0}$, it is implied that $\mathbf{g}(t,\mathbf{y}) = \mathbf{f}(t,\mathbf{y}) = \mathbf{y}'$. Then (3.25) becomes

$$\begin{aligned}
\mathcal{L}_N[\mathbf{y}(t);h] &= \mathcal{L}_L[\mathbf{y}(t);h] \\
&= \left(\sum_{i=0}^{K} \alpha_i\right)\mathbf{y} + \sum_{j=0}^{\infty} \mathbf{C}_j(0)\,\mathbf{g}^{(j)} \\
&= \left(\sum_{i=0}^{K} \alpha_i\right)\mathbf{y} + \sum_{j=0}^{\infty} \mathbf{C}_j(0)\,\mathbf{f}^{(j)} \\
&= \left(\sum_{i=0}^{K} \alpha_i\right)\mathbf{y} + \sum_{j=0}^{\infty} \mathbf{C}_j(0)\,\mathbf{y}^{(j+1)} \\
&= \left(\sum_{i=0}^{K} \alpha_i\right)\mathbf{y} + \mathbf{C}_0(0)\,\mathbf{y}' + \mathbf{C}_1(0)\,\mathbf{y}'' + \ldots + \mathbf{C}_q(0)\,\mathbf{y}^{(q+1)} + \ldots.
\end{aligned}$$

If we reindex the coefficients of $\mathbf{y}^{(j)}$, we get exactly the coefficients of the LMS operator; this confirms that NLMS methods are a generalization of the LMS methods since LMS is a special operator of NLMS. When $\mathbf{A} \neq \mathbf{0}$, the two p's are not compatible since, in this case, the solution depends on an exponential times a polynomial of order p_{NLMS}. Then we have $p_{NLMS} = p_{LMS} - 1$.

3.5 Formulation

3.5.1 General Formula

By our selection, we set { } of (3.21) = **0** for j = 0, 1, . . . , p to give p-th-order methods ,

$$\sum_{i=0}^{K} \alpha_i e^{\mathbf{A}h(K-i)} \left[\frac{\mathbf{I}_i^j(\mathbf{A}h)}{j!}\right] - h \sum_{i=0}^{K} \frac{(ih)^j}{j!} \phi_{Ki}(\mathbf{A}h) = \mathbf{0}. \tag{3.28}$$

The above equation expresses a K-step, p-th-order method that consists of a system of p+1 equations. The choice of K and p will determine whether this system has a unique solution, has many solutions, or has no solution. We make a choice , p=K, which ensures the existence of inverses of **K** and **H**. Thus, we can determine $\phi_{Ki}(\mathbf{A}h)$ based on the choice of α_i's, which can be written as

$$\sum_{i=0}^{K} \alpha_i e^{\mathbf{A}h(K-i)} \left[\frac{\mathbf{A}^{j+1}\mathbf{I}_i^j(\mathbf{A}h)}{j!}\right] = \frac{(\mathbf{A}h)^{j+1}}{j!} \sum_{i=0}^{K} i^j \phi_{Ki}(\mathbf{A}h). \tag{3.29}$$

Substituting (3.23) into (3.29), we get

$$\sum_{i=0}^{K} \alpha_i e^{\mathbf{A}h(K-i)} \left[e^{i\mathbf{A}h} - \sum_{\ell=0}^{j} \frac{(i\mathbf{A}h)^\ell}{\ell!}\right] = \frac{(\mathbf{A}h)^{j+1}}{j!} \sum_{i=0}^{K} i^j \phi_{Ki}(\mathbf{A}h). \tag{3.30}$$

Because of the root condition, (3.30) becomes

$$\sum_{i=0}^{K} \alpha_i e^{\mathbf{A}h(K-i)} \sum_{\ell=0}^{j} \frac{(i\mathbf{A}h)^\ell}{\ell!} = -\frac{(\mathbf{A}h)^{j+1}}{j!} \sum_{i=0}^{K} i^j \phi_{Ki}(\mathbf{A}h). \tag{3.31}$$

If we multiply both sides of (3.30) by $(\mathbf{A}h)^{-(j+1)}$ and let $\|\mathbf{A}\| \to 0$, we get

$$\sum_{i=0}^{K} \alpha_i \frac{i^{j+1}}{(j+1)!} = \sum_{i=0}^{K} \frac{i^j}{j!} \phi_{Ki}(\mathbf{0}). \tag{3.32}$$

Then j = 0 gives

$$\sum_{i=0}^{K} i\alpha_i = \sum_{i=0}^{K} \phi_{Ki}(\mathbf{0}),$$

the consistency condition for LMS methods for $\phi_{Ki}(\mathbf{0}) = \beta_{Ki} = \beta_i$. This confirms that NLMS methods are a generalization of LMS methods.

We determine $\phi_{Ki}(\mathbf{A}h)$, without loss of generality, by selecting $\alpha_K = 1$. In addition, we require the condition of strong stability to be realized, i.e., $\mathrm{Re}\{\lambda(\mathbf{A})\} < 0$. The $\phi_{Ki}(\mathbf{A}h)$ are determined by means of (3.31), which can be considered as a matrix equation for K-step, p-th-order method ($K \geq 1$, $p \geq 0$). On the other hand, we can select $\phi_{Ki}(\mathbf{A}h)$ to determine α_i's as well, but we choose not to do this since we would have to investigate the strong stability of the resulting α_i's. It is easier to choose α_i's to be strongly stable and then solve for $\phi_{Ki}(\mathbf{A}h)$. An approach to the selection of α_i's to ensure strong stability is presented below. The $(K \times K)$ companion matrix of the characteristic polynomial $\rho(\zeta) = \sum_{i=0}^{K} \alpha_i \zeta^i$ takes the form

$$\begin{pmatrix} 0 & 0 & \cdots & 0 & -\alpha_0 \\ 1 & 0 & & 0 & -\alpha_1 \\ 0 & 1 & & 0 & -\alpha_2 \\ \vdots & & & & \vdots \\ 0 & 0 & \cdots & 1 & -\alpha_{K-1} \end{pmatrix}$$

for α_K chosen to be 1. We require that $\rho(1) = 0$; then $\sum_{i=0}^{K} \alpha_i$ must equal 0. Apply Gerschgorin's estimate columnwise to obtain the eigenvalues of the companion matrix, and impose the condition that the eigenvalues lie on the boundary or inside the unit circle. For the first $(K-1)$ columns, we have the same estimate:

$$|\lambda - 0| \leq 1 .$$

The estimate of the last column gives

$$|\lambda - (-\alpha_{K-1})| \leq \sum_{i=0}^{K-2} |\alpha_i| ,$$

which implies that

$$|\lambda| \leq \sum_{i=0}^{K-1} |\alpha_i|$$

where we require the bound to be ≤ 1.

Next, we look for conditions of α_i that produce strong stability. Consider

$$\rho_1(\zeta) = \frac{\rho(\zeta)}{\zeta - 1} = \sum_{i=0}^{K-1} \hat{\alpha}_i \zeta^i ,$$

where

$$\hat{\alpha}_i = \sum_{j=0}^{K} \alpha_j - \sum_{\ell=0}^{i} \alpha_\ell .$$

The associated $(K-1) \times (K-1)$ companion matrix of the characteristic polynomial $\rho_1(\zeta)$ takes the form

$$\begin{pmatrix} 0 & 0 & \cdots & 0 & -\hat{\alpha}_0 \\ 1 & 0 & \cdots & 0 & -\hat{\alpha}_1 \\ 0 & 1 & & 0 & -\hat{\alpha}_2 \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ 0 & 0 & \cdots & 1 & -\hat{\alpha}_{K-2} \end{pmatrix}$$

Again, applying the same procedure used above to estimate the eigenvalues and to ensure that the eigenvalues lie inside the unit circle, we get

$$|\lambda - 0| < 1 \quad \text{and} \quad |\lambda - (-\hat{\alpha}_{K-2})| \leq \sum_{i=0}^{K-3} |\hat{\alpha}_i| ,$$

which implies that

$$|\lambda| \leq \sum_{i=0}^{K-2} |\hat{\alpha}_i| < 1 .$$

Thus, using

$$\sum_{i=0}^{K} \alpha_i = 1$$

$$\sum_{i=0}^{K-1} |\alpha_i| \leq 1$$

and

$$\sum_{i=0}^{K-2} \left| \sum_{j=0}^{K} \alpha_j - \sum_{l=0}^{i} \alpha_i \right| < 1 ,$$

we can select α_i's satisfying the condition of strong stability.

3.5.2. Matrix Formula for ϕ_{Ki}

In (3.31), i is a column index and j is a row index. Let ϕ be a vector of K-1 or K elements whose components are $\phi_{Ki}(\mathbf{A}h)$; i = 0, 1, ... , K-1 or K. Then $\phi_{Ki}(\mathbf{A}h)$ can be determined by the matrix equation

$$\phi = -\mathbf{K}^{-1} \mathbf{H}^{-1} \mathbf{E} \Psi . \qquad (3.33)$$

Each of these symbols is described in the next section. For the system of equations, the above matrix elements are all submatrices.

3.5.3. Explicit Schemes

Equation (3.31), expressed as a predictor, takes the following form when $\phi_{KK}(\mathbf{A}h) = \mathbf{0}$:

$$\sum_{i=0}^{K} \alpha_i e^{\mathbf{A}h(K-i)} \sum_{\ell=0}^{j} \frac{(i\mathbf{A}h)^{\ell}}{\ell!} = -\frac{(\mathbf{A}h)^{j+1}}{j!} \sum_{i=0}^{K-1} i^j \phi_{Ki}(\mathbf{A}h) \tag{3.34}$$

for $j = 0, 1, \ldots, p$. Now, we have

$$\begin{pmatrix} I & I & \cdots & I \\ I & I + \mathbf{A}h & \cdots & I + K\mathbf{A}h \\ \vdots & \vdots & & \vdots \\ I & \sum_{m=0}^{p-1} \frac{(\mathbf{A}h)^m}{m!} & \cdots & \sum_{m=0}^{p-1} \frac{(K\mathbf{A}h)^m}{m!} \end{pmatrix} \begin{pmatrix} \alpha_0 e^{K\mathbf{A}h} \\ \alpha_1 e^{(K-1)\mathbf{A}h} \\ \vdots \\ \alpha_K I \end{pmatrix}$$

$$= -\begin{pmatrix} \frac{\mathbf{A}h}{0!} & & & \\ & \frac{(\mathbf{A}h)^2}{1!} & & \\ & & \ddots & \\ & & & \frac{(\mathbf{A}h)^p}{(p-1)!} \end{pmatrix} \begin{pmatrix} I & I & \cdots & I \\ \mathbf{0} & I & \cdots & (K-1)I \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & I & \cdots & (K-1)^{p-1} I \end{pmatrix} \begin{pmatrix} \phi_{K0} \\ \phi_{K1} \\ \vdots \\ \phi_{K,K-1} \end{pmatrix}. \tag{3.35}$$

In the matrix form, we get

$$\mathbf{E}\Psi = -\mathbf{H}\mathbf{K}\phi . \tag{3.36}$$

Thus, ϕ can be obtained by (3.33). Here,

$$\mathbf{H} \equiv (\mathbf{H}_{ii})$$

$$\mathbf{K} \equiv (\mathbf{K}_{ij})$$

$$\mathbf{E} \equiv (\mathbf{E}_{i\ell})$$

$$\Psi \equiv (\Psi_{\ell,1})$$

$$\phi \equiv (\phi_{j,1}) ,$$

where

$i = 1, \ldots , p$

$j = 1, \ldots , K$

$\ell = 1, \ldots , K+1.$

3.5.4. Implicit Schemes

If we use (3.31) as the formula for a NLMS corrector, for $\phi_{KK}(\mathbf{A}h) \neq \mathbf{0}$, we have

$$\sum_{i=0}^{K} \alpha_i \, e^{\mathbf{A}h(K-i)} \sum_{\ell=0}^{j} \frac{(i\mathbf{A}h)^{\ell}}{\ell!} = -\frac{(\mathbf{A}h)^{j+1}}{j!} \sum_{i=0}^{K} i^j \, \phi_{Ki}(\mathbf{A}h)$$

for $j = 0, 1 \ldots , p$. Now we have

$$\begin{pmatrix} I & I & \cdots & I \\ I & I + \mathbf{A}h & \cdots & I + K\mathbf{A}h \\ \vdots & \vdots & & \vdots \\ I & \sum_{m=0}^{p} \frac{(\mathbf{A}h)^m}{m!} & \cdots & \sum_{m=0}^{p} \frac{(K\mathbf{A}h)^m}{m!} \end{pmatrix} \begin{pmatrix} \alpha_o e^{K\mathbf{A}h} \\ \alpha_1 e^{(K-1)\mathbf{A}h} \\ \vdots \\ \alpha_K I \end{pmatrix}$$

$$= - \begin{pmatrix} \frac{\mathbf{A}h}{0!} & & \bigcirc \\ & \frac{(\mathbf{A}h)^2}{1!} & \\ \bigcirc & & \frac{(\mathbf{A}h)^{p+1}}{p!} \end{pmatrix} \begin{pmatrix} I & I & \cdots & I \\ 0 & I & \cdots & KI \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & I & \cdots & K^p I \end{pmatrix} \begin{pmatrix} \phi_{K0} \\ \phi_{K1} \\ \vdots \\ \phi_{KK} \end{pmatrix} . \tag{3.37}$$

This gives the same matrix form as (3.36), where

$$\mathbf{H} \equiv (\mathbf{H}_{mm})$$

$$\mathbf{K} \equiv (\mathbf{K}_{mm})$$

$$\mathbf{E} \equiv (\mathbf{E}_{m\ell})$$

$$\Psi \equiv (\Psi_{\ell 1})$$

$$\phi \equiv (\phi_{\ell 1}) \ ,$$

where

$$m = 1, \ldots , p + 1$$

$$\ell = 1, \ldots , K + 1 .$$

3.6. Lemmas

We now come to the most important theorem, the convergence theorem. Before this theorem can be proved, three lemmas are needed. The proofs of two lemmas and of the convergence theorem are an extension and modification of the proofs of LMS methods developed by Henrici [16]. We begin by establishing the formula for discretization error in the approximate solution of an arbitrary differential equation by NLMS methods.

3.6.1. Lemma 3.6.1

Consider the general characteristic polynomial,

$$\rho(\lambda,\zeta) = \rho(\lambda(\mathbf{A}),\zeta) = \sum_{i=0}^{K} \alpha_i \left(e^{\lambda hK} \zeta^i\right) ,$$

which satisfies the root condition; i.e.,

$$\rho(\lambda, 1) = 0 \quad \forall \ \lambda = \lambda_j(\mathbf{A}) .$$

It is easily seen that $\rho(\mathbf{A},\zeta)$ also equals $\mathbf{0}$:

$$\mathbf{0} = \rho(\mathbf{A},\zeta) = \sum_{i=0}^{K} \alpha_i \left(e^{\mathbf{A}hK} \zeta^i\right) = \sum_{i=0}^{K} \alpha_i \, e^{\mathbf{A}h(K-i)} \left(e^{\mathbf{A}h} \zeta\right)^i = \mathbf{0} . \qquad (3.38)$$

The following lemma is a generalization of Henrici [16].

Lemma 3.6.1: Define the scalars $\gamma_\ell = 0$ for $\ell < 0$, where ℓ is an integer index. Then, $\exists$ a set of bounded scalars $\{\gamma_\ell\}$ ∋

$$\sum_{i=0}^{K} \alpha_i \, e^{\mathbf{A}h(K-i)} \gamma_{\ell-(K-i)} = \begin{cases} \mathbf{I} ; & \ell = 0 \\ \mathbf{0} ; & \ell > 0 \end{cases} . \qquad (3.39)$$

Proof: Write (3.38) as

$$\rho(\mathbf{A},\zeta) = \rho(\xi) = \sum_{i=0}^{K} \alpha_i e^{\mathbf{A}h(K-i)} \xi^i = \mathbf{0} ,$$

where $\xi = e^{\mathbf{A}h}\zeta$. Consider

$$\rho^{\#}(\xi) = \alpha_K \mathbf{I} + \alpha_{K-1} e^{\mathbf{A}h} \xi + \ldots + \alpha_2 e^{\mathbf{A}h(K-2)} \xi^{K-2} + \alpha_1 e^{\mathbf{A}h(K-1)} \xi^{K-1} + \alpha_o e^{\mathbf{A}hK} \xi^K$$

$$= \xi^K (\alpha_K \xi^{-K} + \alpha_{K-1} e^{\mathbf{A}h} \xi^{-(K-1)} + \ldots + \alpha_2 e^{\mathbf{A}h(K-2)} \xi^{-2} + \alpha_1 e^{\mathbf{A}h(K-1)} \xi^{-1}$$

$$+ \alpha_o e^{\mathbf{A}hK}) = \xi^K \rho(\xi^{-1}) .$$

The roots of each row of $\rho^{\#}(\xi)$ are the reciprocals of each corresponding row of $\rho(\xi)$. Note that $e^{\mathbf{A}h} \neq \mathbf{0}$ and $\rho(\xi) = e^{\mathbf{A}hK} \rho(\zeta)$. By lemma 5.5 (Henrici [16]), the $\rho(\zeta)$ of each row has no roots outside $|\zeta| = 1$ because of the root condition. It follows that $[\rho^{\#}(\xi)]^{-1}$ is holomorphic inside $|\zeta| < 1$ for all n rows. Using the Maclaurin expansion for each row of $[\rho^{\#}(\xi)]^{-1}$, we find

$$[\rho^{\#}(\xi)]^{-1} = \sum_{i=0}^{\infty} \gamma_i \xi^i . \tag{3.40}$$

By Cauchy's estimate, it is seen that all γ_i are bounded. To prove (3.39), we multiply both sides of (3.40) by $\rho^{\#}(\xi)$ and equate the coefficients, obtaining (3.39).

Q.E.D.

From (3.5), since $\alpha_K \neq 0$, we can write

$$\mathbf{y}_{n+K} = \frac{1}{\alpha_K} \left\{ - \sum_{i=0}^{K-1} \alpha_i e^{\mathbf{A}h(K-i)} \mathbf{y}_{n+i} + h \sum_{i=0}^{K} \phi_{Ki}(\mathbf{A}h) \mathbf{g}_{n+i} \right\} , \tag{3.41}$$

which is of the form

$$\mathbf{y} = G(\mathbf{y}) , \tag{3.42}$$

where $\mathbf{y} = \mathbf{y}_{n+K}$. The successive iterative form gives

$$\mathbf{y}^{(\nu+1)} = G\left(\mathbf{y}^{(\nu)}\right) \tag{3.43}$$

for any initial vector $\mathbf{y}^{(0)}$.

Let $G(\mathbf{y})$ be defined for $\|\mathbf{y}\| < \infty$, and let $\exists$ a constant $k \ni 0 \leq k < 1$. Then $G(\mathbf{y})$ satisfies the condition

$$\|G(\mathbf{y}^*) - G(\mathbf{y})\| \leq k \|\mathbf{y}^* - \mathbf{y}\| . \tag{3.44}$$

Using the definition of $G(\mathbf{y})$, formula (3.41), and the fact that $\mathbf{g}(t,\mathbf{y})$ satisfies the Lipschitz condition with Lipschitz constant L, we see that condition (3.44) is satisfied by

$$k = \frac{h \|\phi_{KK}(\mathbf{A}h)\|}{\alpha_K} L \tag{3.45}$$

for sufficiently small h and for all $\|\mathbf{A}\| < \infty$.

For the iterative procedure (3.43) to converge for arbitrary initial $\mathbf{y}^{(0)}$, k is required to be < 1:

$$k < 1 \rightarrow \frac{h \|\phi_{KK}(\mathbf{A}h)\|}{\alpha_K} L < 1 . \tag{3.46}$$

Conventionally, when using LMS K-step methods with $\alpha_K = 1$, we select $h \ni$

$$\beta_K h L^* \sim \kappa \quad (<1) . \tag{3.47}$$

Similarly, for NLMS K-step methods, we select h_N to satisfy condition (3.46):

$$\|\phi_{KK}(\mathbf{A}h_N)\| \; h_N \; L \sim \kappa . \tag{3.48}$$

Combining (3.47) and (3.48), we see that

$$h_N = \left\|\phi_{KK}^{-1}(\mathbf{A}h_N)\right\| \; \beta_K \; \frac{L^*}{L} \; h .$$

For $\left\| \phi_{KK}^{-1}(\mathbf{A}h_N) \right\| \beta_K$ not too small, we know that $L^* >> L$; therefore, $h_N >> h$. This tells us that we can choose a much larger step size using NLMS than we can choose using LMS.

3.6.2. Lemma 3.6.2

Lemma 3.6.2: Let h satisfy the condition (3.46). Then

$$\sum_{i=0}^{K} \| \phi_{Ki}(\mathbf{A}h) \| < \infty .$$

Proof: The $\phi_{Ki}(\mathbf{A}h)$ are linear transformations in a finite-dimensional vector space; therefore, they are completely continuous (Bachman [1]). Every completely continuous transformation is itself continuous. In finite-dimensional spaces, a linear transformation is bounded if and only if it is continuous. Hence,

$$\| \phi_{Ki}(\mathbf{A}h) \| < \infty \tag{3.49}$$

for $i = 0, 1, 2, \ldots, K$; therefore,

$$\sum_{i=0}^{K} \| \phi_{Ki}(\mathbf{A}h) \| < \infty .$$

Q.E.D.

3.6.3. Lemma 3.6.3

The next lemma, a generalization of Henrici [16], concerns the error growth of NLMS methods. The proof of this lemma is preceded by a list of necessary definitions. Let

(1) $\sup_i \{|\gamma_i|\} < \Gamma$

(2) $\sum_{i=0}^{K} |\alpha_i| = a$

(3) $$E = \begin{cases} \max\left|e^{K\lambda(\mathbf{A})\,h}\right|; & 0 < \mathrm{Re}\{\lambda(\mathbf{A})\} < \infty \\ 1 \qquad\qquad\quad ; & \mathrm{Re}\{\lambda(\mathbf{A})\} \leq 0 \end{cases}$$

(4) Let $\mathbf{z}_\nu$ be initial guesses

$$\|\mathbf{z}_\nu\| \leq \mathcal{Z} \;\forall\, \nu$$

(5) Let λ_ν be the growth parameters

$$\|\lambda_\nu\| < \Lambda \;\forall\, \nu$$

(6) $$\sigma = \Gamma \left\| (I - h\,\phi_{K,n-K})^{-1} \right\|$$

(7) $$\sum_{i=0}^{K} \|\phi_{Ki}(\mathbf{A}h)\| \leq \Phi$$

(8) $n = 0, 1, \ldots, N$.

Lemma 3.6.3: The growth of the solution $(\mathbf{z}_m)$ of NLMS satisfies the following inequality:

$$\|\mathbf{z}_n\| \leq \sigma(\mathcal{E}\mathcal{Z} + N\Lambda)\, e^{nh\sigma\Phi}. \tag{3.52}$$

Proof: We begin by examining the error growth of the NLMS nonhomogeneous difference equation:

$$\alpha_K \mathbf{z}_{m+K} + \alpha_{K-1} e^{\mathbf{A}h} \mathbf{z}_{m+K-1} + \ldots + \alpha_o e^{K\mathbf{A}h} \mathbf{z}_m$$

$$= h\left\{\phi_{K,m}\,\mathbf{z}_{m+K} + \phi_{K-1,m}\,\mathbf{z}_{m+K-1} + \ldots \phi_{o,m}\,\mathbf{z}_m\right\} + \lambda_m\,, \tag{3.53}$$

where $m = n - K - \ell$; $\ell = 0, 1, \ldots, n-K$. For $\mathbf{A} = \mathbf{0}$, this reduces to the LMS case. For $\mathbf{A} \neq \mathbf{0}$, we approach an error bound. We multiply both sides of (3.53) by γ_ℓ for $\ell = 0, 1, \ldots, n-K$. We then sum up each side and use formula (3.39) to obtain:

$$\text{LHS} = \mathbf{z}_n + (\alpha_{K-1} e^{\mathbf{A}h} \gamma_{n-K} + \ldots + \alpha_o e^{K\mathbf{A}h} \gamma_{n-2K+1}) \mathbf{z}_{K-1}$$

$$+ (\alpha_{K-2} e^{2\mathbf{A}h} \gamma_{n-K} + \ldots + \alpha_o e^{K\mathbf{A}h} \gamma_{n-2K+2}) \mathbf{z}_{K-2} + \ldots$$

$$+ \alpha_o e^{K\mathbf{A}h} \gamma_{n-K} \mathbf{z}_o$$

and

$$\text{RHS} = h\left\{\phi_{K,n-K} \gamma_o\right\} \mathbf{z}_n + h\left\{\phi_{K-1,n-K} \gamma_o + \phi_{K,n-K-1} \gamma_1\right\} \mathbf{z}_{n-1} + \ldots$$

$$+ h\left\{\phi_{o,n-K} \gamma_o + \phi_{1,n-K-1} \gamma_1 + \ldots + \phi_{K,n-2K} \gamma_K\right\} \mathbf{z}_{n-K} + \ldots$$

$$+ h\left\{\phi_{o,o} \gamma_{n-K}\right\} \mathbf{z}_o + (\lambda_{n-K} \gamma_o + \lambda_{n-K-1} \gamma_1 + \ldots + \lambda_o \gamma_{n-K}) .$$

The coefficients of $\mathbf{z}_o, \mathbf{z}_1, \ldots, \mathbf{z}_{n-1}$ of the RHS are functions of γ_i, $\phi_{i,n-K-i}$.

If we apply lemma 3.6.2, the norm of the sum of each term inside $\{\ \} \leq \phi\Gamma$.

Equating LHS and RHS gives

$$\mathbf{z}_n + (\ ) \mathbf{z}_{K-1} + (\ ) \mathbf{z}_{K-2} + \ldots + (\ ) \mathbf{z}_o = h\phi_{K,n-K} \gamma_o \mathbf{z}_n + h\{\ \} \mathbf{z}_{n-1} + \ldots$$

$$+ h\{\ \} \mathbf{z}_o + (\lambda_{n-K} \gamma_o + \lambda_{n-K-1} \gamma_1 + \ldots + \lambda_o \gamma_{n-K}) .$$

Solving the above equation for $\mathbf{z}_n$, we obtain

$$\mathbf{z}_n = (I - h\phi_{K,n-K} \gamma_o)^{-1} \left[h\{\ \} \mathbf{z}_{n-1} + h\{\ \} \mathbf{z}_{n-2} + \ldots + h\{\ \} \mathbf{z}_o - \left\{(\ ) \mathbf{z}_{K-1}\right.\right.$$

$$\left.\left. + (\ ) \mathbf{z}_{K-2} + \ldots + (\ ) \mathbf{z}_o\right\} + (\lambda_{n-K} \gamma_o + \lambda_{n-K-1} \gamma_1 + \ldots + \lambda_o \gamma_{n-K})\right] . \quad (3.54)$$

Applying our above definitions to the terms of (3.54), we obtain

$$\|(\ ) \mathbf{z}_{K-1} + (\ ) \mathbf{z}_{K-2} + \ldots + (\ ) \mathbf{z}_o\| \leq \left(\sum_{m=0}^{K} |\alpha_m|\right) E\Gamma \quad (3.55)$$

$$\|\lambda_{n-K} \gamma_o + \lambda_{n-K-1} \gamma_1 + \ldots + \lambda_o \gamma_{n-K}\| \leq N\Lambda\Gamma \quad (3.56)$$

$$\|h\{\ \} \mathbf{z}_{n-1} + h\{\ \} \mathbf{z}_{n-2} + \ldots + h\{\ \} \mathbf{z}_o\| \leq h\Gamma\phi \sum_{m=0}^{n-1} \|\mathbf{z}_m\| . \quad (3.57)$$

Then, taking norms of both sides of (3.54) and applying estimates (3.55), (3.56), and (3.57), we find

$$\|\mathbf{z}_n\| \le \|(I - h\phi_{K,n-K}\gamma_0)^{-1}\| \left\{ \Gamma \tfrac{1}{2} G E + \Gamma h \phi \sum_{m=0}^{n-1} \|\mathbf{z}_m\| + \Gamma N \Lambda \right\}$$

$$= \sigma \left\{ h\phi \sum_{m=0}^{n-1} \|\mathbf{z}_m\| + \tfrac{1}{2} G E + N\Lambda \right\} . \tag{3.58}$$

Let $L^{\#} = \sigma\phi$,

$K^{\#} = \sigma(\tfrac{1}{2} G E + N\Lambda)$;

then (3.58) takes the form

$$\|\mathbf{z}_n\| \le h L^{\#} \sum_{m=0}^{n-1} \|\mathbf{z}_m\| + K^{\#} . \tag{3.59}$$

Note that $G E \sigma \ge 1 \rightarrow K^{\#} \ge \tfrac{1}{2}$. Using mathematical induction, we obtain the estimate

$$\|\mathbf{z}_m\| \le K^{\#} (1 + h L^{\#})^m \tag{3.60}$$

true for $m = 0, 1, \ldots, K-1$. Assuming that (3.60) is true for $m = 0, 1, \ldots, n-1$ and using

$$\|\mathbf{z}_n\| \le h L^{\#} \sum_{m=0}^{n-1} \|\mathbf{z}_m\| + K^{\#}$$

and the formula

$$\sum_{j=0}^{n-1} x^j = \frac{x^n - 1}{x - 1} \quad \text{for } x \ne 1 ,$$

we get

$$\|\mathbf{z}_n\| \le h L^{\#} \sum_{m=0}^{n-1} K^{\#} (1 + h L^{\#})^m + K^{\#} = h L^{\#} K^{\#} \frac{(1 + h L^{\#})^n - 1}{h L^{\#}} + K^{\#} = K^{\#} (1 + h L^{\#})^n .$$

Therefore, we establish that

$$\|\mathbf{z}_n\| \leq K^{\#} e^{nhL^{\#}} . \tag{3.61}$$

Substituting the definitions of $K^{\#}$ and $L^{\#}$ into (3.61), we obtain

$$\|\mathbf{z}_n\| \leq \sigma(\|E\|\xi + N\Lambda)\, e^{nh\sigma\phi}.$$

This is exactly the inequality (3.52), which established the truth for $m=n$. Therefore, (3.52) holds generally for $m=0,1,2,\ldots,N$.

Q.E.D.

In section 3.2, Stability, we mentioned that the root condition is a sufficient condition for stability. In this section we show that it is sufficient.

We let $\hat{\zeta}$ denote the maximum of the moduli of the roots of the characteristic polynomial $\rho(\zeta)$. Let the first K initial vectors, $\mathbf{y}_n$, satisfy

$$\|\mathbf{y}_n\| \leq \hat{\zeta}^n \xi \quad \text{for } n = 0,1,2,\ldots,K-1,$$

where ξ is a constant. Let $\mathbf{z}$ be a set of starting vectors whose starting values satisfy

$$\|\mathbf{z}_n\| \leq \xi .$$

If we set $\mathbf{z}_n = \hat{\zeta}^{-n}\mathbf{y}_n$ and apply lemma 3.6.3 to $\mathbf{z}_n$, we find

$$\|\mathbf{y}_n\| \leq \hat{\zeta}^n \sigma(\|E\|\xi + N\Lambda)\, e^{nh\sigma\phi}.$$

Since all roots lie on or inside the unit circle, $\|\mathbf{y}_n\|$ remains bounded, thus establishing the stability. This completes the proof of the following stability theorem.

Stability Theorem: A nonlinear multistep method is strongly stable if and only if its characteristic polynomial satisfies the strong root condition.

Now, we proceed, by utilizing all the available lemmas and definitions, to prove the convergence theorem.

3.7 Convergence Theorem

Theorem 3.7: (Convergence Theorem)

A strongly stable and consistent NLMS method is convergent.

The first step in our approach is to estimate the $\mathcal{L}_N[\mathbf{y}(t);h]$ at $t = t_n$. In our general formula (3.5), $\mathbf{g}(t,\mathbf{y})$ is assumed to belong to C^{p+1}, in which case we can directly use (3.25) to estimate the nonlinear operator. However, $\mathbf{g}(t,\mathbf{y})$ may not always be differentiable, and, in this case, we need to use a different approach to estimate the nonlinear operator, which is what we will do in our proof. We want to use the condition for stability and consistency to prove that

$$\lim_{\substack{h\to 0 \\ t=t_n}} \mathbf{y}_n = \mathbf{y}(t) \quad \forall \; t \in [a,b].$$

Note that the use of strong stability gives a desirable estimate for $\mathcal{L}_N[\mathbf{y}(t);h]$ since $\mathrm{Re}\{\lambda(\mathbf{A})\} < 0$. However, growth estimates may differ, depending on whether $\mathrm{Re}\{\lambda(\mathbf{A})\} < 0$ or $\mathrm{Re}\{\lambda(\mathbf{A})\} \geq 0$.

Proof: Let $\mathbf{y}(t)$ be the solution of $\mathbf{y}' = \mathbf{A}\mathbf{y} + \mathbf{g}(t,\mathbf{y})$; $\mathbf{y}(t_o) = \mathbf{y}_o$

$\mathbf{y}_n$ be the solution of $\sum_{i=0}^{K} \alpha_i e^{\mathbf{A}h(K-i)} \mathbf{y}_{n+i} = h \sum_{i=0}^{K} \phi_{Ki} \mathbf{g}_{n+i}$

$\mathbf{y}_{jo}$ be the starting values for $j = 0, 1, \ldots, K-1$.

Set

$$\delta(h) = \max_j \|\mathbf{y}_{jo} - \mathbf{y}(t_o + jh)\|,$$

and assume that $\lim_{h\to 0} \delta(h) = 0$. We want to show that for any $t \in [t_o, b]$,

$$\lim_{\substack{h\to 0 \\ t_n=t}} \mathbf{y}_n = \mathbf{y}(t).$$

Before we come to the proof, let us make use of both the stability and the consistency conditions to derive an identity that will be used to estimate the NLMS operator $\mathcal{L}_N[\mathbf{y}(t);h]$. Formula (3.31) gives

$$\sum_{i=0}^{K} \alpha_i \, e^{\mathbf{A}h(K-i)} \sum_{\ell=0}^{j} \frac{(i\mathbf{A}h)^\ell}{\ell!} = -\frac{(\mathbf{A}h)^{j+1}}{j!} \sum_{i=0}^{K} i^j \, \phi_{Ki}(\mathbf{A}h) \, .$$

Set $j = 0$. Simplifying the above formula, we obtain

$$\sum_{i=0}^{K} \alpha_i \, e^{\mathbf{A}h(K-i)} + \mathbf{A}h \sum_{i=0}^{K} \phi_{Ki}(\mathbf{A}h) = \mathbf{0} \, . \tag{3.62}$$

This is a consistency condition when $j = 0$.

Define for $\epsilon \geq 0$, the M. O. C. (modulus of continuity):

$$\omega(\epsilon) = \max_{\substack{|t^*-t| \leq \epsilon \\ t^*, t \in [t_o, t]}} \|\mathbf{g}(t^*, \mathbf{y}) - \mathbf{g}(t, \mathbf{y})\| \, .$$

For $i = 0, 1, \ldots, K$, we can write

$$\mathbf{g}(t, \mathbf{y}) = \mathbf{g}(t_n, \mathbf{y}) + \theta_i^g \, \omega(ih) \, , \tag{3.63}$$

where

$$\|\theta_i^g\| \leq 1 \text{ and } |t - t_n| \leq ih \, .$$

Substituting (3.63) into (3.2), we obtain

$$\mathbf{y}(t_{n+i}) = e^{i\mathbf{A}h} \mathbf{y}(t_n) + \int_{t_n}^{t_{n+i}} e^{\mathbf{A}(t_{n+i}-t')} [\mathbf{g}(t_n, \mathbf{y}) + \theta_i^g \, \omega(ih)] \, dt' \, .$$

Then,

$$\mathbf{y}(t_{n+i}) = e^{i\mathbf{A}h} \mathbf{y}(t_n) + \int_{t_n}^{t_{n+i}} e^{\mathbf{A}(t_{n+i}-t')} \, dt' \; \mathbf{g}(t_n, \mathbf{y}) + \int_{t_n}^{t_{n+i}} e^{\mathbf{A}(t_{n+i}-t')} \, dt' \cdot \theta_i^g \, \omega(ih) \, . \tag{3.64}$$

If we apply

$$\int_{t_n}^{t_{n+i}} e^{\mathbf{A}(t_{n+i}-t')} \, dt' = -\mathbf{A}^{-1} \{I - e^{i\mathbf{A}h}\} \, ,$$

then (3.64) becomes

$$\mathbf{y}(t_{n+i}) = e^{i\mathbf{A}h}\,\mathbf{y}(t_n) - \mathbf{A}^{-1}\{I - e^{i\mathbf{A}h}\}\,\mathbf{g}(t_n,\mathbf{y}) - \mathbf{A}^{-1}\{I - e^{i\mathbf{A}h}\}\,\theta_i^g\,\omega(ih)\,.$$

If we write $\mathbf{A}^{-1}\{I - e^{i\mathbf{A}h}\} = O(h)$, the above formula becomes

$$\mathbf{y}(t_{n+i}) = e^{i\mathbf{A}h}\,\mathbf{y}(t_n) - \mathbf{A}^{-1}\{I - e^{i\mathbf{A}h}\}\,\mathbf{g}(t_n,\mathbf{y}) - O(h)\,\theta_i^g\,\omega(ih)\,.$$

Multiplying both sides by $\alpha_i\, e^{\mathbf{A}h(K-i)}$ and summing over i, we get

$$\sum_{i=0}^{K}\alpha_i\, e^{\mathbf{A}h(K-i)}\,\mathbf{y}(t_{n+i}) = \left\{\sum_{i=0}^{K}\alpha_i\, e^{\mathbf{A}h(K-i)}\, e^{i\mathbf{A}h}\,\mathbf{y}(t_n)\right\}$$

$$-\left[\mathbf{A}^{-1}\sum_{i=0}^{K}\alpha_i\, e^{\mathbf{A}h(K-i)}\,(I - e^{i\mathbf{A}h})\,\mathbf{g}(t_n,\mathbf{y})\right] - \sum_{i=0}^{K}\alpha_i\, e^{\mathbf{A}h(K-i)}\,O(h)\,\theta_i^g\,\omega(ih)\,. \qquad (3.65)$$

Since the method is stable, it must satisfy the root condition. Therefore, $\rho(1) = \sum_{i=0}^{K}\alpha_i = 0$, which implies $\{\ \}$ of (3.65) $= e^{\mathbf{A}hK}\,\mathbf{y}(t_n)\sum_{i=0}^{K}\alpha_i = \mathbf{0}$. Simplifying $[\ ]$ of (3.65) and applying the same root condition, we find

$$-\mathbf{A}^{-1}\sum_{i=0}^{K}\alpha_i\, e^{\mathbf{A}h(K-i)}\,(I - e^{i\mathbf{A}h})\,\mathbf{g}(t_n,\mathbf{y}) = -\mathbf{A}^{-1}\sum_{i=0}^{K}\alpha_i\, e^{\mathbf{A}h(K-i)}\,\mathbf{g}(t_n,\mathbf{y})\,.$$

Therefore,

$$\sum_{i=0}^{K}\alpha_i\, e^{\mathbf{A}h(K-i)}\,\mathbf{y}(t_{n+i}) = -\mathbf{A}^{-1}\sum_{i=0}^{K}\alpha_i\, e^{\mathbf{A}h(K-i)}\,\mathbf{g}(t_n,\mathbf{y})$$

$$-\,O(h)\sum_{i=0}^{K}\alpha_i\, e^{\mathbf{A}h(K-i)}\,\theta_i^g\,\omega(ih)\,. \qquad (3.66)$$

And

$$h\sum_{i=0}^{K}\phi_{Ki}(\mathbf{A}h)\,\mathbf{g}(t_{n+i},\mathbf{y}) = h\sum_{i=0}^{K}\phi_{Ki}(\mathbf{A}h)\,\mathbf{g}(t_n,\mathbf{y}) + h\sum_{i=0}^{K}\phi_{Ki}(\mathbf{A}h)\,\theta_i^g\,\omega(ih)\,. \qquad (3.67)$$

Then, (3.66) - (3.67) gives

$$\mathcal{L}_N[\mathbf{y}(t);h] = \left\{-\mathbf{A}^{-1}\left(\sum_{i=0}^{K}\alpha_i\, e^{\mathbf{A}h(K-i)} + \mathbf{A}\,h\sum_{i=0}^{K}\phi_{Ki}(\mathbf{A}h)\right)\mathbf{g}(t_n,\mathbf{y})\right\}$$

$$-\sum_{i=0}^{K}\left(O(h)\,\alpha_i\, e^{\mathbf{A}h(K-i)} + h\phi_{Ki}(\mathbf{A}h)\right)\theta_i^{g}\,\omega(ih)\,.$$

By the consistency condition (3.62), $\{\ \} \to 0$.

$$\therefore\ \|\mathcal{L}_N[\mathbf{y}(t);h]\| \le Q^{\#}\,h\,\omega(ih)\,,$$

where

$$Q^{\#}\,h = \sum_{i=0}^{K}\left(\|O(h)\|\,|\alpha_i|\,\|e^{\mathbf{A}h(K-i)}\| + h\,\|\phi_{Ki}(\mathbf{A}h)\|\right).$$

Formula (3.5) - $\mathcal{L}_N[\mathbf{y}(t);h]$ gives

$$\sum_{i=0}^{K}\alpha_i\, e^{\mathbf{A}h(K-i)}\left(\mathbf{y}_{n+i} - \mathbf{y}(t_{n+i})\right) - h\sum_{i=0}^{K}\phi_{Ki}\left(\mathbf{g}_{n+i} - \mathbf{g}(t_{n+i},\mathbf{y}(t_{n+i}))\right) = \mathbf{Q}\theta_i^{g}\,h\,\omega(ih)\,,$$

where

$$\mathbf{Q} = \sum_{i=0}^{K}\left(O(h)\,\alpha_i\, e^{\mathbf{A}h(K-i)} + h\,\phi_{Ki}(\mathbf{A}h)\right)\,.$$

Let $\mathbf{y}_n - \mathbf{y}(t_n) = \mathbf{e}_n$. In view of the Lipschitz condition for $\mathbf{g}$, i.e.,

$$\|\mathbf{g}(t_n,\mathbf{y}(t_n)) - \mathbf{g}(t_n,\mathbf{y}_n)\| \le L\,\|\mathbf{y}(t_n) - \mathbf{y}_n\|\,,$$

we can define

$$\mathbf{g}(t_n,\mathbf{y}(t_n)) - \mathbf{g}(t_n,\mathbf{y}_n) = \begin{cases} \bar{\mathbf{g}}_n\,\mathbf{e}_n & \text{for } \|\mathbf{e}_n\| \neq 0 \\ \mathbf{0} & \text{for } \|\mathbf{e}_n\| = 0 \end{cases}\,,$$

so that we get

$$\sum_{i=0}^{K}\alpha_i\, e^{\mathbf{A}h(K-i)}\,\mathbf{e}_{n+i} - h\sum_{i=0}^{K}(\phi_{Ki})\ \bar{\mathbf{g}}_{n+i}\,\mathbf{e}_{n+i} = \mathbf{Q}\theta_i^{g}\,h\,\omega(Kh)\,.$$

We now apply lemma 3.6.3 with

$$\mathbf{e}_n = \mathbf{z}_n ; \quad \delta(h) = ?$$

$$\mathcal{A} = |\alpha_0| + |\alpha_1| + \ldots + |\alpha_K|$$

$$E = \max_i \|e^{i\mathbf{A}h}\|$$

$$\sum_{i=0}^{K} \|\phi_{Ki}\| \bar{g}_{n+i} \leq \sum_{i=0}^{K} \|\phi_{Ki}\| \quad L = L\phi$$

$$\sigma = \|(I - h\phi_{KK}(\mathbf{A}h))^{-1}\|$$

$$N = \frac{t_n - t_o}{h}$$

$$\Lambda = \|\mathbf{Q}\| \, h\omega(Kh)$$

and obtain

$$\|\mathbf{e}_n\| \leq \sigma\left(\mathcal{A} E\delta(h) + (t_n - t_o)\|\mathbf{Q}\| \, \omega(Kh)\right)e^{(t_n - t_o)\sigma L\phi}$$

as $h \to 0$ and both $\delta(h)$ and $\omega(Kh) \to 0$.

$\therefore$ the above bound $\mathbf{e}_n \to 0$ for every $t \in [t_o, b]$, establishing the results.

Q. E. D.

4. COMPUTATIONAL CONSIDERATIONS

In this section, we first provide a general description of the different types of computational errors and then describe how we treat them at the present test stage. Then we introduce an algorithm to compute $e^{\mathbf{A}}$ when $\mathbf{A}$ is a function of t.

We define the error function $\mathbf{C}_{p+1}$ to be the first term of the initial local discretization error of the NLMS operator. We perform a general analysis on the error function $\mathbf{C}_{p+1}$, which is dependent on the characteristic polynomial coefficients and the characteristic roots. We will show that, for LMS methods, it is possible to select values of α_i such that $\mathbf{C}_{p+1}$ reaches a minimum. We will show, by example, that these methods are not strongly stable. Some interesting results are presented, which although they are not made conclusive at this time, do provide information for future research.

4.1 General Considerations

Errors in computation by NLMS methods are attributable to the following sources:

(1) Input and output conversion errors
(2) Computational round-off errors
(3) Matrix inversion errors
(4) Pade approximation errors
(5) Local and global discretization errors.

Errors of types (1) and (2) depend on the computing device and the software package. Accuracy can be maintained at a desired level by using double-precision arithmetic.

To minimize the errors, the matrix inversion package developed by Forsythe [12] was used for all test problems. If necessary, these errors can be improved further by using double-precision arithmetic.

The Pade approximation is stable as a consequence of the Pade lemma, section 3.2.2.

The bounds of the local discretization errors can be estimated by formula (3.52). The error function $\mathbf{C}_{p+1}$ will be discussed independently in section 4.3. Since NLMS methods are strongly stable when applied to the solutions of stiff equations, the global discretization errors remain bounded within the numerical approximation provided by the NLMS methods. The growth of this type of error for all $\mathbf{A}$ can be estimated by applying lemma 3.6.3.

The round-off errors depend on the precision of the computing device, which in turn is dependent on the number of digits used. This type of error is also dependent on the number of operations involved. It should be noted that when applying NLMS methods, the computation of $e^{\mathbf{A}}$ and the inversion of $(\mathbf{A}h)^K$ could have been carried out by techniques other than those used here. To aid in the appraisal of the round-off errors incurred when obtaining $\mathbf{y}_{n+1}$, we have provided table 1 to show the number of operations required for each iteration in terms of scalar, vector, and matrix operations. As far as operational counts are concerned, the use of NLMS methods does not result in fewer operations than does LMS methods. Albeit the LMS methods are not optimal in this sense, a submember of the LMS methods, i.e., the Adams family, does minimize the number of arithmetic operations because it chooses $\alpha_K = 1$, $\alpha_{K-1} = -1$, and the remaining α's = 0. Of course, where functions have operations in common,

the operations should be calculated beforehand so as to minimize the total number of calculations.

Table 1. Table of Operations

Description	Step 1				Step 2				Step 3			
	Explicit		Implicit		Explicit		Implicit		Explicit		Implicit	
	I	S	I	S	I	S	I	S	I	S	I	S
(Scalar) (Vector)	1		1		1		1		1		1	
(Scalar) (Matrix)	2		5		4		7		10		35	
Vector Addition	1	1	2	2	3	3	4	4	5	5	6	6
Matrix Addition	1		2		4		15		22		46	
(Matrix) (Vector)	2	2	3	3	4	4	5	5	6	6	7	7
Matrix Multiplication	1		4		5		11		14		19	
Pade	1	*	1	*	2	*	2	*	3	*	3	*
Matrix Inversion	1	*	1	*	1	*	1	*	1	*	1	*
Evaluation of $\mathbf{g}(t,\mathbf{y})$	1	1	2	2	2	2	3	3	3	3	4	4

Symbols: I - Initial step
S - Subsequent steps
* - Needed if $\mathbf{A} = \mathbf{A}(t)$

4.2 Function $\mathbf{A}(t)$

For $\mathrm{Re}\{\lambda(\mathbf{A}(t))\} < 0 \ \forall \ t$, $\mathbf{A}(t)$ can be evaluated by the Pade approximation at every t_i. Lawson [24] introduced an Algorithm II, a periodic estimation of

$$\mathbf{A} = \mathbf{A}_k = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right|_{\substack{t = kh \\ \mathbf{y} = \mathbf{y}(t)}},$$

which can be adopted in our evaluation of $\mathbf{A}(t)$.

This thesis takes $\mathbf{A}(t)$ into consideration, but in the test examples $\mathbf{A}$ is chosen to be a real constant matrix. The evaluation of $\mathbf{A}(t)$ will be included in the computation package, which is already under development by the author.

4.3 The Error Function $\mathbf{C}_{p+1}$

We will now examine the error function $\mathbf{C}_{p+1}$ for the LMS methods of order p. The following three questions arise:

(1) Can we select α_i among the strongly stable families such that $\mathbf{C}_{p+1}$ is at its smallest magnitude?

(2) If the answer is "yes" to question (1), is the Adams family the optimal family?

(3) If the answer is "no" to question (1), what is the choice for α_i such that $\mathbf{C}_{p+1}$ is at its smallest magnitude? Which family exhibits this characteristic?

When answering the above questions, one should note that when $\mathbf{A} = \mathbf{0}$, NLMS methods reduce to LMS methods. If we multiply both sides of formula (3.30) by $[(\mathbf{A}h)^{j+1}]^{-1}$ and let $\|\mathbf{A}\| \to 0$, we get

$$\sum_{i=0}^{K} \alpha_i \frac{i^{j+1}}{(j+1)!} = \frac{1}{j!} \sum_{i=0}^{K} i^j \phi_{Ki}(0) . \tag{4.1}$$

For $j = 0, 1, \ldots, p-1$, equation (4.1) gives a formulation of the LMS methods of order p. The matrix form is:

$$\begin{pmatrix} 0 & 1 & \cdots & K \\ 0 & \frac{1^2}{2!} & \cdots & \frac{K^2}{2!} \\ \vdots & \vdots & & \vdots \\ 0 & \frac{1^p}{p!} & \cdots & \frac{K^p}{p!} \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_K \end{pmatrix} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 0 & 1 & \cdots & K \\ \vdots & \vdots & & \vdots \\ 0 & \frac{1^{p-1}}{(p-1)!} & \cdots & \frac{K^{p-1}}{(p-1)!} \end{pmatrix} \begin{pmatrix} \phi_{K0} \\ \phi_{K1} \\ \vdots \\ \phi_{KK} \end{pmatrix} . \tag{4.2}$$

We are interested in expressing β_i ($= \phi_{Ki}(0)$) as a function of α_i, and α_i as a function of the characteristic roots r_j. Thus we need to perturb the roots to determine the effect on the error function C_{p+1} when LMS methods are used, particularly when using the Adams family of the LMS methods. For convenience in using formula (4.2), let us list the first few C_{p+1}'s for LMS methods of order up to 3.

Explicit Integrator:

For $K = 1$, $p = 1$,

$$\phi_{1,0} = \alpha_1$$

$$\text{LMS} \quad C_2 = \frac{1}{2}\,\alpha_1$$

$$\text{Adams} \quad C_2 = \frac{1}{2}\,.$$

For $K = 2$, $p = 2$,

$$\begin{pmatrix} \phi_{2,0} \\ \phi_{2,1} \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\,\alpha_1 \\ \frac{1}{2}\,\alpha_1 + 2\alpha_2 \end{pmatrix}$$

$$\text{LMS} \quad C_3 = -\frac{1}{12}\,\alpha_1 + \frac{1}{3}\,\alpha_2$$

$$\text{Adams} \quad C_3 = \frac{5}{12}\,.$$

For K = 3, p = 3,

$$\begin{pmatrix} \phi_{3,0} \\ \phi_{3,1} \\ \phi_{3,2} \end{pmatrix} = \begin{pmatrix} \frac{5}{12}\ \alpha_1 + \frac{1}{3}\ \alpha_2 + \frac{3}{4}\ \alpha_3 \\ \frac{8}{12}\ \alpha_1 + \frac{4}{3}\ \alpha_2 \\ -\frac{1}{12}\ \alpha_1 + \frac{1}{3}\ \alpha_2 + \frac{9}{4}\ \alpha_3 \end{pmatrix}$$

LMS $\quad C_4 = \frac{1}{24}\ \alpha_1 + \frac{9}{24}\ \alpha_3$

Adams $\quad C_4 = \frac{9}{24}$.

Implicit Integrator:

For K = 1, p = 2,

$$\begin{pmatrix} \phi_{1,0} \\ \phi_{1,1} \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\ \alpha_1 \\ \frac{1}{2}\ \alpha_1 \end{pmatrix}$$

LMS $\quad C_3 = -\frac{1}{12}\ \alpha_1$

Adams $\quad C_3 = \frac{1}{12}$.

For K = 2, p = 3,

$$\begin{pmatrix} \phi_{2,0} \\ \phi_{2,1} \\ \phi_{2,2} \end{pmatrix} = \begin{pmatrix} \frac{5}{12}\ \alpha_1 + \frac{1}{3}\ \alpha_2 \\ \frac{8}{12}\ \alpha_1 + \frac{4}{3}\ \alpha_2 \\ -\frac{1}{12}\ \alpha_1 + \frac{1}{3}\ \alpha_2 \end{pmatrix}$$

LMS $\quad C_4 = \frac{1}{24}\ \alpha_1$

Adams $\quad C_4 = -\frac{1}{24}$.

For K = 3, p = 4,

$$\begin{pmatrix} \phi_{3,0} \\ \phi_{3,1} \\ \phi_{3,2} \\ \phi_{3,3} \end{pmatrix} = \begin{pmatrix} \frac{9}{24}\alpha_1 + \frac{1}{3}\alpha_2 + \frac{3}{8}\alpha_3 \\ \frac{19}{24}\alpha_1 + \frac{4}{3}\alpha_2 + \frac{9}{8}\alpha_3 \\ -\frac{5}{24}\alpha_1 + \frac{1}{3}\alpha_2 + \frac{9}{8}\alpha_3 \\ \frac{1}{24}\alpha_1 + \frac{3}{8}\alpha_3 \end{pmatrix}$$

LMS $\quad C_5 = -\frac{19}{720}\alpha_1 - \frac{1}{90}\alpha_2 - \frac{3}{80}\alpha_3$

Adams $\quad C_5 = -\frac{19}{720}$

.
.
.

etc.

The resulting C_{p+1} in terms of α_i is summarized in table 2.

Let ζ_j be the roots of the characteristic polynomial $\rho(\zeta)$ satisfying the root condition of stability. The relationship

$$\prod_{j=1}^{K} (\zeta - \zeta_j) = \sum_{i=0}^{K} \alpha_i \zeta^i ; \quad \alpha_K = 1 \tag{4.3}$$

enables us to express α_i as a function of ζ_j. From (4.2) we can express $\phi_{Ki}(0)$ as a function of α_i; $\phi_{Ki}(0)$ can also be expressed as a function of ζ_j

according to formula (4.3). Using $\zeta_1 = 1$, we can convert table 2 into table 3, which describes C_{p+1} as a function of ζ_j.

Table 2. C_{p+1} in Terms of α_i for LMS Methods

	Step K \ Order p	C_2	C_3	C_4	C_5
Predictor	1	$\frac{1}{2}\alpha_1$			
	2		$-\frac{1}{12}\alpha_1 + \frac{1}{3}\alpha_2$		
	3			$\frac{1}{24}\alpha_1 + \frac{9}{24}\alpha_3$	
	4				$-\frac{19}{720}\alpha_1 - \frac{1}{90}\alpha_2$ $-\frac{3}{80}\alpha_3 + \frac{14}{45}\alpha_4$
Corrector	1		$-\frac{1}{12}\alpha_1$		
	2			$\frac{1}{24}\alpha_1$	
	3				$-\frac{19}{720}\alpha_1 - \frac{1}{90}\alpha_2$ $-\frac{3}{80}\alpha_3$
	4				

Table 3. C_{p+1} in Terms of ζ_j for LMS Methods

	Step K \ Order p	C_2	C_3	C_4	C_5
Predictor	1	$\frac{1}{2}$			
	2		$\frac{5}{12}+\frac{1}{12}\zeta_2$		
	3			$\frac{9}{24}+\frac{1}{24}(\zeta_2+\zeta_3)$ $+\frac{1}{24}\zeta_2\zeta_3$	
	4				$\frac{19}{720}(\zeta_2+\zeta_3+\zeta_4)$ $+\frac{11}{720}(\zeta_2\zeta_3+\zeta_{2\ 4}+\zeta_3\zeta_4)$ $+\frac{19}{720}\zeta_2\zeta_3\zeta_4+\frac{251}{720}$
Corrector	1		$-\frac{1}{12}$		
	2			$-\frac{1}{24}-\frac{1}{24}\zeta_2$	
	3				$-\frac{19}{720}-\frac{11}{720}(\zeta_2+\zeta_3)$ $-\frac{11}{720}\zeta_2\zeta_3$
	4				

Using table 3, we can examine the change in C_{p+1} when we perturb the root ζ_j.

Initially, let us examine the second-order predictor,

$$\text{LMS} \quad C_3 = \frac{5}{12} + \frac{1}{12}\,\zeta_2 .$$

The choice of $\zeta_2 = 0$ implies that Adams $C_3 = \frac{5}{12}$.

In order to perturb the root ζ_2, we write

$$\text{LMS } C_3 = \frac{5}{12} + \frac{1}{12}\,(\zeta_2 + \epsilon) .$$

It is easily seen that when $\zeta_2 + \epsilon = -1$, C_3 will be at a minimum. We vary ϵ to perturb ζ_2. When we keep ϵ positively small, the method remains in the strongly stable family. As $\epsilon \to 0$ and $\zeta_2 \to -1$, this method is shifted to a weakly stable family. Now we have found an interesting answer to question (2): Among LMS methods of order p, the Adams family does not have the smallest error C_{p+1}. Similarly, from the same example, we observe that LMS methods of order p that possess the smallest error C_{p+1} are not strongly stable. This can serve as an answer to question (1). We now arrive at the following conjecture: Among LMS methods of order p, the weakly stable families possess the smallest error C_{p+1}.

The above study indicates that there should exist an NLMS family that possesses the smallest error function C_{p+1}. This optimal family is not yet identified and will not be identified in this thesis.

We desire to make the best possible choice of α_i, so that when applying NLMS methods, a minimum error function C_{p+1} results. Our recommendation is

as follows: Identify an NLMS family whose $\alpha_K = 1$, $\alpha_{K-1} = -1$, $\alpha_{K-2} = \alpha_{K-3} = \ldots = \alpha_o = 0$. This family, which is a generalization of the Adams family, we label GA. We refer to the predictors of this family as GAB (Generalized Adams Bashforth) and the correctors of this family as GAM (Generalized Adams Moulton).

We have made several numerical investigations with our test problems on different selections of α_i, from both strongly and weakly stable LMS families, without noting much difference in the results. Without a thorough round-off error analysis, one cannot tell how to choose α_i's that will reduce the error function C_{p+1}. However, as pointed out earlier, the Adams family has the least number of operations, and this will aid in the reduction of round-off errors. Until a thorough analysis is made of C_{p+1}, we recommend the GA family as the representative family for all the strongly stable NLMS methods.

5. NUMERICAL COMPARISONS

Hull [19] and Ehle [11] have thoroughly tested selected existing numerical methods for solving initial value problems in ordinary differential equations; Hull tested for both stiff and nonstiff differential equations, and Ehle tested for stiff differential equations. Their work stressed the requirement that in order to compare methods, meaningful criteria must be defined. Since the tests included in this thesis are limited, we do not need an extensive rule for testing purposes.

There are two basic reasons why we do not intend to perform extensive tests:

(1) At this stage, the principal objective is to confirm whether NLMS methods work effectively on the selected problems. Some features, such as PEC^m and double-precision arithmetic, are not yet incorporated in our present computation package.

(2) Although some of the answers that can be obtained by using the Hull-Ehle test criteria would be desirable, those answers are not required for our present purposes.

However, we do define reasonable test criteria that should result in a meaningful comparison of NLMS methods against the selected methods. Since each problem is different in nature, we will define specific test criteria for each problem.

To test NLMS methods, we developed a program package for use with the Univac 1108. It is written in FORTRAN V language and in single-precision arithmetic. Adams' formulas are written in the same language and are inserted in the program wherever needed. Gear's program is run independently.

NLMS methods have been tested extensively on a number of stiff problems. Five stiff problems whose solutions are known have been selected for presentation. Problem 5, selected from Ehle's report, has four complex eigenvalues; NLMS methods did very well on this problem. Problem 6 is presented to demonstrate how well an NLMS method can handle a nonstiff problem; this problem has one eigenvalue whose real part >0. The results of the comparisons are presented by graph or table or both, depending on the need in each case.

5.1 Problem 1

Problem Description:

$$y' = -100y + (1 + t^2); \quad y(0) = 1.$$

Exact Solution:

$$y(t) = \left(1 - \frac{1}{100} - \frac{2}{100^3}\right) e^{-100t} + \frac{1}{100} + \frac{1}{100^3} (100^2 t^2 - 200t + 2).$$

Problem Parameters:

Time Interval: $[0, 1.95]$

Step Size h : $h = \frac{1}{2^i}$; $i = 1, 2, \ldots, 14$.

NLMS Methods Applied:

Explicit 2-step.

Compare Against:

Gear's program and second-order Adams-Bashforth method.

Comparison Criteria:

(1) Relative error is used as the measure of success when comparing methods.

(2) Gear's program, Adams' method, and NLMS-2-step are used with the same fixed $h = 2^{-14}$ over $t \in [0, 1.95]$.

(3) Adams' method and NLMS-2-step are used with different $h = \frac{1}{2^i}$, $i = 1, 2, \ldots, 14$ over $t \in [0, 0.3]$, where the exponential makes a

significant contribution. For a larger step size > 0.3, where the exponential term damps out, the comparison is made on five successive computations.

Description of Comparisons:

Table 4: Comparison Among Methods of Gear, Adams, and NLMS of Order 2 with Fixed $h = 2^{-14}$.

Table 5: Comparison Between Methods of Adams and NLMS of Order 2 for Different h.

Figure 1: t versus $\text{Log}_{10} E$.

Figure 2: $-\text{Log}_2 h$ versus $\text{Log}_{10} E$.

Eigenvalues: -100.

Source: S. Preiser (1969).

Remarks:

Gear's variable-order technique, as applied to this problem, involves trying different orders up to order 3. However, most computations are carried out with a second-order stiff method with an acceptable initial $h = 2^{-14}$.

Figure 1 shows that for a fixed small step size, the nonlinear multi-2-step method produces more accurate results in terms of relative error.

Figure 2 shows that to maintain an accuracy of the order of 10^{-6}, Adams' methods require a step size of 2^{-14}, whereas the nonlinear multi-2-step method can use a step size of 2^{-9} to maintain the same accuracy.

Table 4. Comparison Among Methods of Gear (G), Adams (A), and NLMS (N) of Order 2 with Fixed $h = 2^{-14}$

Number of Steps $t = nh$	Method	Relative Error
1	G	.2163 8476 E-03
	A	.1055 7626 E-06
	N	.0
500	G	.7845 4426 E-02
	A	.4139 7907 E-04
	N	.3764 9440 E-05
1,000	G	.3423 0498 E-02
	A	.1871 0187 E-04
	N	.1504 4324 E-05
5,000	G	.8566 0224 E-06
	A	.8565 7385 E-06
	N	.9636 4558 E-07
10,000	G	.6845 3327 E-06
	A	.6844 9640 E-06
	N	.4620 3507 E-06
15,000	G	.1061 7635 E-05
	A	.1061 6991 E-05
	N	.1534 9866 E-06
20,000	G	.7836 8658 E-06
	A	.7836 3961 E-06
	N	.3398 9188 E-06
25,000	G	.1115 4447 E-05
	A	.1115 3821 E-05
	N	.3953 2531 E-06
30,000	G	.8522 7857 E-06
	A	.8522 3467 E-06
	N	.3236 3342 E-07

Table 5. Comparison Between Methods of Adams (A) and NLMS (N) of Order 2 for Differenc h

$-\text{Log}_2$ h	Method	Relative Error
14	A	.4944 3871 E-06
	N	.2317 6814 E-07
13	A	.3797 0580 E-05
	N	.5607 4696 E-07
12	A	.3049 6416 E-04
	N	.4306 1870 E-07
11	A	.2472 3962 E-03
	N	.5971 7310 E-07
10	A	.2028 3254 E-02
	N	.5312 1861 E-07
9	A	.1685 8731 E-01
	N	.7057 9671 E-07
8	A	.1383 0503 E-00
	N	.1135 2181 E-05
7	A	.1269 6386 E+01
	N	.3016 8420 E-04
6	A	.3223 7045 E+03
	N	.2619 8035 E-03
5	A	.3333 8325 E+05
	N	.1237 5420 E-02
4	A	.1516 9195 E+07
	N	.5426 6854 E-02
3	A	.4206 1372 E+08
	N	.1789 7609 E-01
2	A	.7013 3530 E+09
	N	.3655 0522 E-01
1	A	.7568 7259 E+10
	N	.4881 1901 E-01

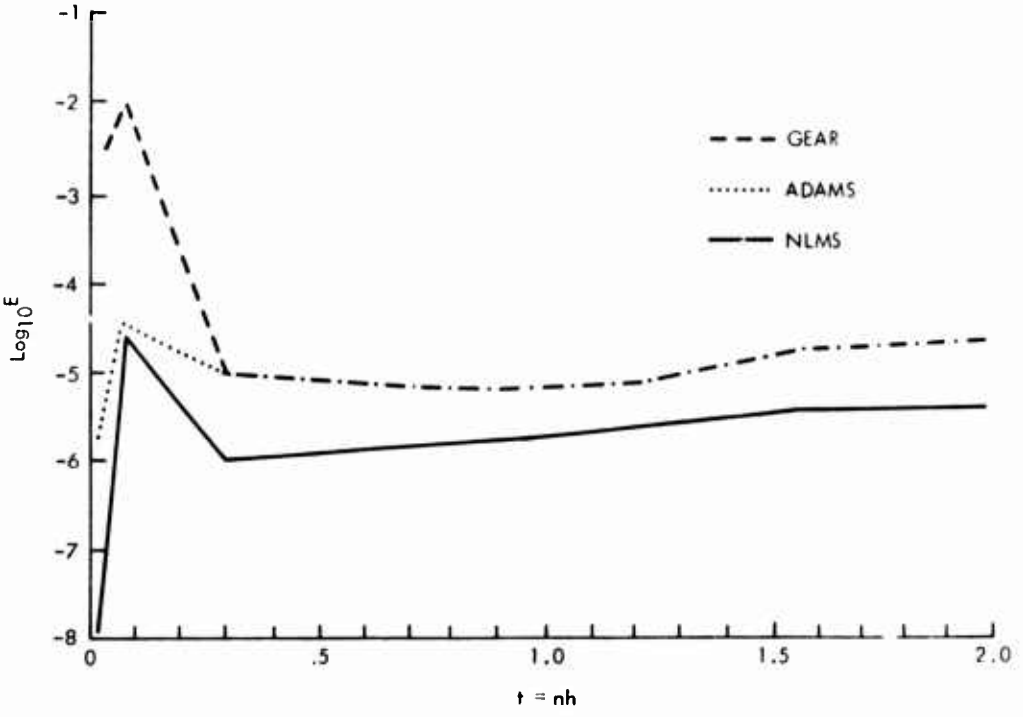


Figure 1. t versus $Log_{10}E$

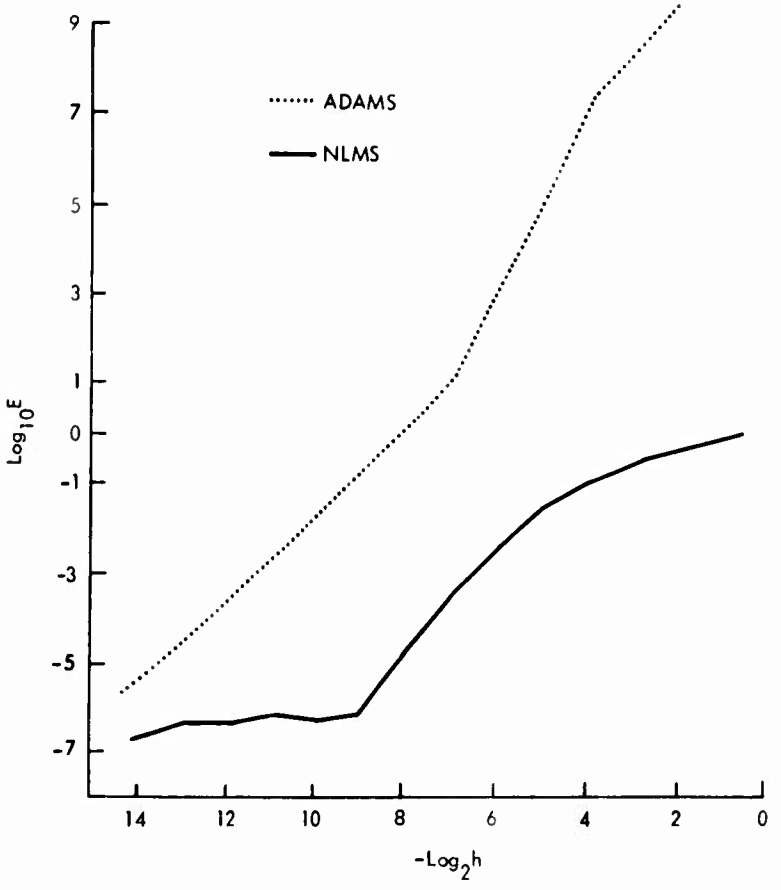


Figure 2. $-Log_2 h$ versus $Log_{10}E$

5.2 Problem 2

Problem Description:

$$y' = -100y + (1 + t^2 - t^4); \quad y(0) = 1 .$$

Exact Solution:

$$y(t) = \left(\frac{99}{100} - \frac{2}{100^3} + \frac{24}{100^5}\right) e^{-100t} + \left(\frac{1}{100} + \frac{2}{100^3} - \frac{24}{100^5}\right)$$

$$+ \left(\frac{-2}{100^2} + \frac{24}{100^4}\right) t + \left(\frac{1}{100} - \frac{12}{100^3}\right) t^2 + \frac{4}{100^2} t^3 - \frac{1}{100} t^4 .$$

Problem Parameters:

Time Interval: $[0, 20]$

Step Size h : $2^N(.1E\text{-}05)$; $N = 0, 1, \ldots .$

NLMS Methods Applied:

Explicit 2-step.

Compare Against:

Second-order Adams-Moulton method.

Comparison Criteria:

Compare the results, by means of relative errors, after the first calculation since the initial local discretization and round-off errors are the smallest then.

Description of Comparisons:

Table 6: Comparison Between Methods of NLMS-2-Step and Second-Order Adams Moulton.

Figure 3: $\text{Log}_{10} E$ versus N.

Eigenvalues: -100.

Source: S. Preiser (1969).

Remarks:

Figure 3 shows that, for a required accuracy of 10^{-7}, Adams' step size needs to be chosen $\sim 2^{6}$ (.2 E-06); NLMS can maintain the same accuracy using a step size $\sim 2^{13}$ (.2 E-06). This shows that $h_N = 100 h_A = \|\mathbf{A}\| h_A$, confirming our analysis on the step size.

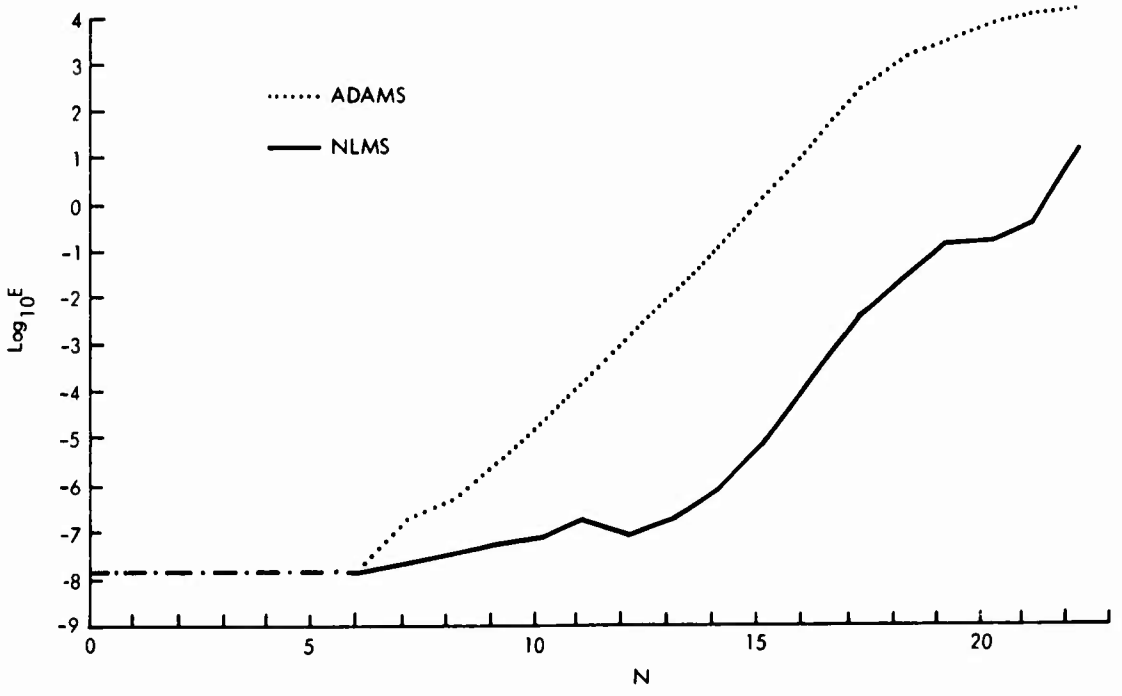


Figure 3: $\text{Log}_{10} E$ versus N

Table 6. Comparison Between Methods of NLMS-2-Step and Second-Order Adams Moulton

$h = 2^N (.1E-06)$	Relative Error	
N	NLMS Explicit 2	Adams Bashforth 2
0	.0	.7450 8757 E-08
1	.1490 2341 E-07	.0
2	.7451 7609 E-08	.7451 7609 E-08
3	.7452 9413 E-08	.7452 9413 E-08
4	.7455 3027 E-08	.7455 3027 E-08
5	.1492 0056 E-07	.0
6	.1493 8974 E-07	.0
7	.0	.2246 5323 E-07
8	.1505 2984 E-07	.5268 5443 E-07
9	.7603 1685 E-08	.4561 9011 E-06
10	.7758 8486 E-08	.3623 3823 E-05
11	.1615 9463 E-07	.2963 6455 E-04
12	.0	.2478 3312 E-03
13	.1029 9568 E-07	.2167 8635 E-02
14	.7108 4830 E-07	.2076 5414 E-01
15	.7129 6095 E-06	.2390 2048 E-00
16	.1606 4611 E-04	.3890 9526 E+01
17	.5174 4164 E-03	.7090 2860 E+02
18	.4007 2017 E-02	.2529 4871 E+03
19	.1700 8785 E-01	.4998 2459 E+03
20	.5041 3123 E-01	.9109 8973 E+03
21	.5908 2495 E-01	.1709 4020 E+04
22	.1357 5798 E+01	.1050 7818 E+04

5.3 Problem 3

Problem Description:

$$\mathbf{y}' = \begin{pmatrix} -1 & 95 \\ -1 & -97 \end{pmatrix}\mathbf{y} \; ; \quad \mathbf{y}(0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} .$$

Exact Solution.

$$\mathbf{y}(t) = \frac{1}{47}\begin{pmatrix} 95\,e^{-2t} - 48\,e^{-96t} \\ 48\,e^{-96t} - e^{-2t} \end{pmatrix} .$$

Problem Parameters:

Time Interval: $[0, 10]$

Step Size h : $h = \frac{10}{2^i}$; $i = 0, 1, \ldots, 15$.

NLMS Methods Applied:

Explicit methods of steps 1, 2, and 3.

Compare Against:

Exact solution.

Comparison Criteria:

The tolerance definition follows Ehle; i.e.,

$$\text{ymax}_i = \max\left\{\|\mathbf{y}_0\|, \max|y_k|\right\} ; \quad k = 0, 1, \ldots, i$$

$$\text{Error} = \left\| \frac{\mathbf{y}_{i+1} - \mathbf{y}(t_{i+1}, t_i, \mathbf{y}_i)}{\text{ymax}_i} \right\|_2$$

Ehle's tolerance = 10^{-7} .

Results are tabulated at $t = 10$ for different h by different NLMS methods in terms of above error.

Description of Comparisons:

Table 7: Ehle Errors by NLMS Methods for Different h

(Max. Ehle error = $10^{-4.8}$ to $10^{-5.9}$).

Eigenvalues: $\{-2, -96\}$.

Source: Ehle initial value problem 1 [11].

Table 7. Ehle Errors by NLMS Methods for Different h
(Max. Ehle error = $10^{-4.8}$ to $10^{-5.9}$)

$h = \frac{10}{2^i}$ i	Error		
	1-Step	2-Step	3-Step
0	.1622 8589 E-11		
1	.1622 8981 E-11	.7368 3619 E-16	
2	.1622 8589 E-11	.1217 2031 E-11	.8113 5093 E-12
3	.1622 8981 E-11	.1420 1095 E-11	.1217 2423 E-11
4	.1622 9374 E-11	.1521 5431 E-11	.1420 2273 E-11
5	.1622 3879 E-11	.1571 7888 E-11	.1521 3075 E-11
6	.1552 2089 E-11	.1597 9716 E-11	.1572 6132 E-11
7	.1327 9476 E-11	.1540 0083 E-11	.1597 5790 E-11
8	.1244 0601 E-11	.1323 2902 E-11	.1424 9808 E-11
9	.1284 1213 E-11	.1241 2022 E-11	.1274 2658 E-11
10	.1451 7434 E-11	.1283 5312 E-11	.1242 0126 E-11
11	.1637 2653 E-11	.1462 6200 E-11	.1350 7056 E-11
12	.1646 6864 E-11	.1646 5294 E-11	.1565 7965 E-11
13	.1682 9968 E-11	.1682 0547 E-11	.1682 0547 E-11
14	.1757 1092 E-11	.1757 1092 E-11	.1756 2456 E-11
15	.1857 5613 E-11	.1857 5613 E-11	.1857 5613 E-11

5.4 Problem 4

Problem Description:

$$\mathbf{y}' = \begin{pmatrix} -0.1 & -49.9 & 0 \\ 0 & -50 & 0 \\ 0 & 70 & -120 \end{pmatrix} \mathbf{y}\,; \quad \mathbf{y}(0) = \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix} .$$

Exact Solution:

$$\mathbf{y}(t) = \begin{pmatrix} e^{-0.1t} + e^{-50t} \\ e^{-50t} \\ e^{-120t} + e^{-50t} \end{pmatrix} .$$

Problem Parameters:

Time Interval: $[0, 10]$

Step Size h : 0.01 and 0.2.

NLMS Methods Applied:

Explicit methods of steps 1, 2, and 3.

Compare Against:

Trapezoidal rule.

Comparison Criteria:

(1) The error definition follows Seinfeld; i.e.,

$$\text{Error } R = \frac{y_n - y(t_n)}{y(t_n)} \quad \text{for each component.}$$

(2) The computation time is also tabulated.

Description of Comparisons:

Table 8: Relative Error Comparison Between NLMS Methods and Trapezoidal Rule.

Eigenvalues: $\{-0.1, -50, -120\}$.

Source: Seinfeld, Lapidus, and Hwang [29].

Remarks:

Note that NLMS methods of different steps produce errors of the same order of magnitude. This is expected because we designed the methods to solve the problem $\mathbf{y}' = \mathbf{A}\mathbf{y}$ effectively.

Table 8. Relative Error Comparison Between NLMS Methods and Trapezoidal Rule

Method	h	R_1		R_2	R_3	Time (sec)
		t = 0.4	t = 10	t = 0.4	t = 0.4	
Trapezoidal Rule	.20	1.0×10^{-3}	2.7×10^{-4}	6.5×10^{7}	1.3×10^{5}	1.3
NLMS-1-Step	.01	1.9×10^{-5}	4.7×10^{-4}	2.3×10^{-6}	2.5×10^{-6}	≤ 1
NLMS-2-Step	.01	1.9×10^{-5}	4.7×10^{-4}	2.3×10^{-6}	2.6×10^{-6}	≤ 1
NLMS-3-Step	.01	1.9×10^{-5}	4.7×10^{-4}	2.4×10^{-6}	2.6×10^{-6}	≤ 2
NLMS-1-Step	.20	1.9×10^{-5}	4.5×10^{-4}	4.0×10^{-6}	4.0×10^{-6}	≤ 1

5.5 Problem 5

Problem Description:

$$\mathbf{y}' = \begin{pmatrix} -10^4 & 10^3 & 0 & 0 \\ -10^3 & -10^4 & 0 & 0 \\ 0 & 0 & -50 & 10 \\ 0 & 0 & -10 & -50 \end{pmatrix} \mathbf{y} \; ; \quad \mathbf{y}(0) = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} .$$

Exact Solution:

$$y_1(t) = e^{-10^4 t}\left[\cos(10^3 t) + \sin(10^3 t)\right]$$
$$y_2(t) = e^{-10^4 t}\left[\cos(10^3 t) - \sin(10^3 t)\right]$$
$$y_3(t) = e^{-50 t}\left[\cos(10 t) + \sin(10 t)\right]$$
$$y_4(t) = e^{-50 t}\left[\cos(10 t) - \sin(10 t)\right].$$

Problem Parameters:

Time Interval: $[0, 3]$

Step Size h : $3/2^{\ell}$; $\ell = 0, 1, \ldots, 15$.

NLMS Methods Applied:

Explicit method of step 1.

Compare Against:

Exact solution.

Comparison Criteria:

The tolerance definition follows Ehle; see problem 3. Results are tabulated at t = .91552374 E-01 for different h in terms of above error.

Description of Comparisons:

Table 9: Largest Ehle Errors by NLMS-1-Step for Different h

(Max. Ehle error = $10^{-2.6}$ to $10^{-3.1}$) .

Eigenvalues: $\{-50 \pm 10i,\ -10^4 \pm 10^3 i\}$.

Source: Ehle initial value problem 3 [11].

Remarks:

Note that table 9 indicates increased errors with decreased step sizes; this is probably due to round-off errors. In addition, it is not surprising that, for the largest step sizes, errors become zero since NLMS methods were designed to solve this type of problem exactly in the absence of round-off errors.

Table 9. Largest Ehle Errors by NLMS-1-Step for Different h
(Max. Ehle error = $10^{-2.6}$ to $10^{-3.1}$)

$h = \frac{3}{2^i}$	Error
i	1-Step
0	0
1	0
2	0
3	.8092 0713 E-16
4	.3215 5494 E-12
5	.1028 9758 E-09
6	.2478 1019 E-09
7	.1472 5503 E-08
8	.8835 3017 E-08
9	.8011 5398 E-08
10	.3071 9531 E-07
11	.3071 9531 E-07
12	.3950 8640 E-07
13	.2070 1878 E-06
14	.7513 3183 E-07
15	.4879 6726 E-06

5.6 Problem 6

Problem Description:

$$\mathbf{y}' = \begin{pmatrix} 0 & 1 \\ 10 & -9 \end{pmatrix}\mathbf{y} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \; ; \; \mathbf{y}(0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} .$$

Exact Solution:

$$\mathbf{y}(t) = \begin{pmatrix} 2e^{t} - 1 \\ 2e^{t} - 1 \end{pmatrix} .$$

Problem Parameters:

Time Interval: $[0, 0.082]$

Step Size h : $1/2^{8}$.

NLMS Methods Applied:

Explicit NLMS-2-step method.

Compare Against:

Exact solution.

Comparison Criteria:

Using vector $\|\bullet\|_{\infty}$, compare results against exact solutions.

Description of Comparisons:

Table 10: Table of Numerical Results and Exact Solutions.

Eigenvalues: $\{-10, 1\}$.

Source: S. Preiser (1969).

Remarks:

NLMS methods were designed to be effective for $\mathbf{g}(t, \mathbf{y})$ belonging to the class of slowly varying, low-order polynomials; therefore, it is not surprising that the NLMS-2-step method produces accurate results.

Table 10. Table of Numerical Results and Exact Solutions
(N: NLMS-2-Step; T: Exact Solution)

t	Method	Solution		Relative Error
		$y_1(t)$	$y_2(t)$	
.7812 5000 E-02	N	.1015 6862 E+01	.1015 6862 E+01	.1467 1029 E-07
	T	.1015 6862 E+01	.1015 6862 E+01	
.2343 7500 E-01	N	.1047 4286 E+01	.1047 4286 E+01	0
	T	.1047 4286 E+01	.1047 4286 E+01	
.4296 8750 E-01	N	.1087 8105 E+01	.1087 8106 E+01	.1369 8305 E-07
	T	.1087 8105 E+01	.1087 8105 E+01	
.6250 0000 E-01	N	.1128 9889 E+01	.1128 9889 E+01	.2639 7356 E-07
	T	.1128 9889 E+01	.1128 9889 E+01	
.8203 1250 E-01	N	.1170 9794 E+01	.1170 9794 E+01	.5090 1528 E-07
	T	.1170 9795 E+01	.1170 9795 E+01	

6. FUTURE RELATED RESEARCH

The following areas of future research are desirable:

(1) Develop a package of NLMS computer programs with the following features:

(a) The inclusion of LMS methods

(b) Freedom to select α_i

(c) Built-in PC^m procedure with variable step size

(d) Double-precision option

(e) Evaluation of $\mathbf{A}(t)$ at $t = t_i$ and application of Pade approximation to $\mathbf{A}(t_i)$.

With feature (a), NLMS methods can solve nonstiff equations.

With feature (e), NLMS methods can solve nonlinear equations.

(2) Perform a thorough round-off error analysis for NLMS methods.

(3) Develop theory to appraise the error function $\mathbf{C}_{p+1}$ so that for a certain choice of the combination of characteristic roots, one can minimize the error function $\mathbf{C}_{p+1}$.

7. CONCLUSIONS

We have developed a family of strongly and asymptotically stable NLMS methods for solving stiff equations. The advantages of NLMS methods have already been demonstrated in various sections of this thesis. The following conclusions can be drawn from the work in the preceding sections:

(1) Round-off error consideration is absent in this thesis. While it is possible that round-off errors will be large for some problems, all test results produced by NLMS methods using single-precision arithmetic seem negligibly affected by the round-off error encountered.

(2) It is theoretically true that on a per-step basis, higher order methods give better accuracy. However, it appears that low-order NLMS methods (which require fewer steps) maintain adequate accuracy, so that we believe it is not necessary to employ high-order NLMS methods. However, if desired, high-order NLMS methods can be derived easily. The use of high-order NLMS methods can increase computation time and round-off error and, in addition, would require an accurate Pade approximation and matrix inversion.

(3) The findings from the analysis of $\mathbf{C}_{p+1}$ present some preliminary information for developing criteria for selecting the characteristic roots needed to minimize $\mathbf{C}_{p+1}$. Dahlquist [10] mentioned that, for a certain class of equations, it would be desirable to have an A-stable method with a small truncation error. Once the minimization of $\mathbf{C}_{p+1}$ is obtained for NLMS methods, the NLMS methods seem to provide this.

(4) We have demonstrated that NLMS methods avoid the use of small step size when solving stiff equations. Not only are low-order NLMS methods good predictors and correctors, they can also be used as starters.

(5) After he performed a series of tests on a set of selected stiff equations with a set of existing stiff methods, Ehle [11] concluded that none of the methods used was, by itself, suitable for solving the entire collection of the selected problems. Means have not yet been developed for NLMS methods to handle $\mathbf{A}(t)$; otherwise, NLMS methods have been applied to solve some of Ehle's problems without a single failure. The NLMS methods consistently produced acceptably accurate results. It is felt that when the $\mathbf{A}(t)$ feature is included, NLMS methods will be able to handle a much wider class of stiff problems. At this stage, we can say that NLMS methods are effective for equations whose solutions are asymptotically stable. Indeed, for equations of the form $\frac{d\mathbf{Y}}{dt} = \mathbf{A}\mathbf{y} + \mathcal{P}_n(t)$ (where $\mathcal{P}_n(t)$ is a low-order polynomial in t) in the absence of round-off errors, the technique is exact even for non-asymptotically stable ordinary differential equations.

8. APPENDIX — UNIVAC 1108 FORTRAN V COMPUTER PROGRAMS

This section lists the computer programs, written by the author, that are used to perform numerical experiments by NLMS methods. The programs are written in FORTRAN V language, in independent subroutines of the CALL type, for use on the Univac 1108 computer. Compilation was done by the EXEC 8 system. At the present time, numerical calculations are performed in single-precision arithmetic.

The variable-step-size technique was not applied in testing the selected stiff problems. The technique will, however, be incorporated in the computation package. This package will have an executive program to control the variable step size and will call the listed subroutines as required.

```
C **** NONLINEAR MULTI-K-STEP METHODS
      SUBROUTINE NLMSX(KSTEP,HSIZE,Y,NORD,ALPHA,A,YN,T,INDEX,IS,HOLD)
C **** INDEX=0 CALLS FOR PREDICTOR, OTHERWISE, CORRECTOR
      DIMENSION Y(4,35),ALPHA(1),A(35,35),YN(1),T(1)
      DIMENSION QHI(4,35,35),PHI(4,35,35),P1(35,35)
      DIMENSION UNIT(35,35),AH(35,35),AH2(35,35),AH3(35,35),AH4(35,35)
      DIMENSION EAH(35,35),E2AH(35,35),E3AH(35,35)
      IF(HSIZE-HOLD) 11,10,11
   10 IF(IS .GT. 1) GO TO (100,200,300), KSTEP
   11 CONTINUE
      DO 1 I=1,NORD
      DO 2 J=1,NORD
      P1(I,J)=0.0
      UNIT(I,J)=0.0
      EAH(I,J)=0.0
      E2AH(I,J)=0.0
      E3AH(I,J)=0.0
      AH2(I,J)=0.0
      AH3(I,J)=0.0
      AH4(I,J)=0.0
    2 CONTINUE
      UNIT(I,I)=1.0
      IF(INDEX .EQ. 0) AH3(I,I)=1.0
      AH4(I,I)=1.0
    1 CONTINUE
      DO 3 I=1,4
      DO 4 J=1,NORD
      DO 5 K=1,NORD
      PHI(I,J,K)=0.0
      QHI(I,J,K)=0.0
    5 CONTINUE
    4 CONTINUE
    3 CONTINUE
      DO 6 I=1,NORD
      DO 7 J=1,NORD
      AH(I,J)=HSIZE*A(I,J)
    7 CONTINUE
    6 CONTINUE
      GO TO (100,200,300) , KSTEP
C **** MULTU-1-STEP
  100 CONTINUE
      CALL STEP1(A,NORD,KSTEP,HSIZE,Y,UNIT,P1,AH,EAH,PHI,ALPHA,YN,INDEX,
     $T,IS,HOLD)
      RETURN
C **** MULTI-2-STEP
  200 CONTINUE
      CALL STEP2(A,NORD,KSTEP,HSIZE,Y,UNIT,P1,AH,AH2,EAH,E2AH,PHI,ALPHA,
     $YN,INDEX,T,IS,HOLD)
      RETURN
C **** MULTI-3-STEP
  300 CONTINUE
      CALL STEP3(A,NORD,KSTEP,HSIZE,Y,UNIT,AH,AH2,AH3,AH4,EAH,E2AH,E3AH,
     $PHI,QHI,ALPHA,YN,INDEX,T,IS,HOLD)
      RETURN
      END
```

NONLINEAR MULTI-1-STEP

```
      SUBROUTINE STEP1(A,N,KSTEP,H,Y,UNIT,P1,AH,EAH,PHI,ALPHA,YN,INDEX,T
     $,IS,HOLD)
      DIMENSION P1(35,35),AH(35,35),EAH(35,35),UNIT(35,35),Y(4,35)
      DIMENSION A(35,35),PHI(4,35,35),G(35),ALPHA(1),YN(1),T(1)
      DIMENSION AH2(35,35),E2AH(35,35)
      DO 132 I=1,N
      DO 133 J=1,N
      P1(I,J)=0.0
  133 CONTINUE
      IF(IS .EQ. 1) P1(I,I)=1.0
      YN(I)=0.0
      PHI(2,I,1)=0.0
      PHI(3,I,1)=0.0
  132 CONTINUE
      IF(H-HOLD) 141,140,141
  140 IF(IS.GT.1 .AND. INDEX.EQ.0) GO TO 131
      IF(IS.GT.1 .AND. INDEX.EQ.1) GO TO 170
  141 CONTINUE
C **** P1=(AH) INVERSE
      IF(N -1) 121,120,121
  120 P1(1,1)=1./AH(1,1)
      GO TO 122
  121 CALL INVERT(AH,N,P1)
  122 CONTINUE
C **** EAH=PADE(EXP(AH))
      IF(N -1) 124,123,124
  123 EAH(1,1)=EXP(A(1,1)*H)
      GO TO 125
  124 CALL PADE(A,H,EAH,N)
  125 CONTINUE
      DO 103 I=1,N
      DO 104 J=1,N
C **** AH(I,J)=ALPHA-0*EXP(AH)+I
      AH(I,J)=ALPHA(1)*EAH(I,J)+UNIT(I,J)
  104 CONTINUE
  103 CONTINUE
C **** PREDICTOR
      IF(INDEX .NE. 0) GO TO 150
C **** TO COMPUTE PHI(1,0)
      DO 105 I=1,N
      DO 106 J=1,N
      DO 107 K=1,N
      PHI(1,I,J)=PHI(1,I,J)-P1(I,K)*AH(K,J)
  107 CONTINUE
  106 CONTINUE
  105 CONTINUE
  131 CALL GFN(G,H,N,Y,KSTEP,T,A)
C **** Y(N+1)=YN
      DO 108 I=1,N
      DO 109 J=1,N
      YN(I)=YN(I)+H*PHI(1,I,J)*G(J)
      YN(1)=YN(1)-ALPHA(1)*EAH(I,J)*Y(1,J)
  109 CONTINUE
  108 CONTINUE
```

```
      RETURN
C **** CORRECTOR
C **** TO COMPUTE PHI(1,0) $ (1,1)
  150 CONTINUE
      IF(N - 1) 152,151,152
  151 AH2(1,1)=P1(1,1)*P1(1,1)
      GO TO 153
  152 DO 154 I=1,N
      DO 155 J=1,N
      DO 156 K=1,N
      AH2(I,J)=AH2(I,J)+P1(I,K)*P1(K,J)
  156 CONTINUE
  155 CONTINUE
  154 CONTINUE
  153 CONTINUE
      DO 160 I=1,N
      DO 161 J=1,N
      DO 162 K=1,N
      PHI(1,I,J)=PHI(1,I,J)+ALPHA(1)*H*A(I,K)*EAH(K,J)
  162 CONTINUE
      PHI(1,I,J)=PHI(1,I,J)-AH(I,J)
      E2AH(I,J)=AH(I,J)+H*A(I,J)
  161 CONTINUE
  160 CONTINUE
  170 K2=KSTEP+1
      DO 163 I=1,N
      DO 164 K=1,K2
      CALL GFN(G,H,N,Y,K,T,A)
      DO 165 J=1,N
      IF(K .EQ. 1) PHI(3,I,1)=PHI(3,I,1)+PHI(1,I,J)*G(J)
      IF(K .EQ. 2) PHI(2,I,1)=PHI(2,I,1)+E2AH(I,J)*G(J)
  165 CONTINUE
      PHI(3,I,1)=PHI(3,I,1)+PHI(2,I,1)
  164 CONTINUE
  163 CONTINUE
      DO 166 I=1,N
      DO 167 K=1,N
      YN(I)=YN(I)-H*AH2(I,K)*PHI(3,K,1)-ALPHA(1)*EAH(I,K)*Y(1,K)
  167 CONTINUE
  166 CONTINUE
      RETURN
      END
```

NONLINEAR MULTI-2-STEP

```
      SUBROUTINE STEP2(A,N,KSTEP,H,Y,UNIT,P1,AH,AH2,EAH,E2AH,PHI,ALPHA,Y
     $N,INDEX,T,IS,HOLD)
      DIMENSION P1(35,35),AH(35,35),AH2(35,35),EAH(35,35),E2AH(35,35)
      DIMENSION UNIT(35,35),Y(4,35),A(35,35),PHI(4,35,35),ALPHA(1)
      DIMENSION W(3,3,35,35),AH3(35,35),G(35),YN(1),T(1),QHI(4,35,35)
      DIMENSION E3AH(35,35)
      DO 240 I=1,N
      DO 241 J=1,N
      P1(I,J)=0.0
  241 CONTINUE
      YN(I)=0.0
      PHI(4,20,I)=0.0
      IF(IS .EQ. 1) P1(I,I)=1.0
  240 CONTINUE
      IF(IS .GT. 1) GO TO 212
C **** EAH=PADE(EXP(AH))
C **** E2AH=PADE(EXP(2AH))
      IF(N -1) 206,207,208
  207 EAH(1,1)=EXP(AH(1,1))
      E2AH(1,1)=EAH(1,1)*EAH(1,1)
      GO TO 212
  208 CALL PADE(A,H,EAH,N)
      H2=H+H
      CALL PADE(A,H2,E2AH,N)
  212 CONTINUE
      IF(INDEX .GT. 0) GO TO 260
C **** PREDICTOR
      IF(H-HOLD) 251,250,251
  250 IF(IS .GT. 1) GO TO 234
  251 CONTINUE
C **** AH2(I,J)=(AH) INVERSE SQUARE
      IF(N - 1) 202,201,202
  201 AH2(1,1)=1./AH(1,1)**2
      GO TO 206
  202 CALL INVERT(AH,N,P1)
      DO 203 I=1,N
      DO 204 J=1,N
      DO 205 K=1,N
      AH2(I,J)=AH2(I,J)+P1(I,K)*P1(K,J)
  205 CONTINUE
  204 CONTINUE
  203 CONTINUE
  206 CONTINUE
C **** EAH=PADE(EXP(AH))
C **** COMPUTE PHI(2,0), (2,1)
      DO 213 I=1,N
      DO 214 J=1,N
      P1(I,J)=ALPHA(1)*E2AH(I,J)+ALPHA(2)*EAH(I,J)+UNIT(I,J)
  214 CONTINUE
  213 CONTINUE
      DO 215 I=1,N
      DO 216 J=1,N
      DO 217 K=1,N
      PHI(1,I,J)=PHI(1,I,J)+ALPHA(1)*AH(I,K)*E2AH(K,J)
```

```
      PHI(2,I,J)=PHI(2,I,J)+ALPHA(2)*AH(I,K)*EAH(K,J)
  217 CONTINUE
  216 CONTINUE
  215 CONTINUE
      DO 218 I=1,N
      DO 219 J=1,N
      PHI(1,I,J)=PHI(1,I,J)-P1(I,J)-AH(I,J)
      PHI(2,I,J)=PHI(2,I,J)+P1(I,J)+2.*AH(I,J)
  219 CONTINUE
      P1(1,I)=0.0
  218 CONTINUE
  234 CONTINUE
      DO 220 K=1,KSTEP
      CALL GFN(G,H,N,Y,K,T,A)
      DO 221 I=1,N
      DO 222 J=1,N
      YN(I)=YN(I)+PHI(K,I,J)*G(J)*H
  222 CONTINUE
  221 CONTINUE
  220 CONTINUE
      DO 230 I=1,N
      DO 231 J=1,N
      P1(1,I)=P1(1,I)+AH2(I,J)*YN(J)
  231 CONTINUE
  230 CONTINUE
      DO 232 I=1,N
      DO 233 J=1,N
      PHI(4,20,I)=PHI(4,20,I)-ALPHA(2)*EAH(I,J)*Y(2,J)-ALPHA(1)*E2AH(I,J
     $)*Y(1,J)
  233 CONTINUE
      YN(I)=P1(1,I)+PHI(4,20,I)
  232 CONTINUE
      RETURN
C **** CORRECTOR
  260 CONTINUE
      IF(IS .GT. 1) GO TO 284
      IF(N -1) 262,261,262
  261 AH2(1,1)=AH(1,1)*AH(1,1)
      AH3(1,1)=1./(AH2(1,1)*AH(1,1))
      GO TO 263
  262 DO 264 I=1,N
      DO 265 J=1,N
      DO 266 K=1,N
      AH2(I,J)=AH2(I,J)+AH(I,K)*AH(K,J)
  266 CONTINUE
  265 CONTINUE
  264 CONTINUE
      DO 267 I=1,N
      DO 268 J=1,N
      DO 269 K=1,N
      E3AH(I,J)=E3AH(I,J)+AH2(I,K)*AH(K,J)
  269 CONTINUE
  268 CONTINUE
  267 CONTINUE
      CALL INVERT(E3AH,N,AH3)
  263 CONTINUE
  284 IF(IS .GT. 1) GO TO 279
C **** COMPUTE PHI(2,0), (2,1) $ (2,2)
```

```
      DO 270 I=1,N
      DO 271 J=1,N
      W(1,1,I,J)=UNIT(I,J)-1.5*AH(I,J)+AH2(I,J)
      W(1,2,I,J)=UNIT(I,J)-0.5*AH(I,J)
      W(1,3,I,J)=UNIT(I,J)+0.5*AH(I,J)
      W(2,1,I,J)=-2.*(UNIT(I,J)-AH(I,J))
      W(2,2,I,J)=-2.*UNIT(I,J)+AH2(I,J)
      W(2,3,I,J)=-2.*(UNIT(I,J)+AH(I,J))
      W(3,1,I,J)=W(1,2,I,J)
      W(3,2,I,J)=W(1,3,I,J)
      W(3,3,I,J)=UNIT(I,J)+1.5*AH(I,J)+AH2(I,J)
  271 CONTINUE
  270 CONTINUE
      DO 283 K=1,3
      DO 272 I=1,N
      DO 273 J=1,N
      DO 274 L=1,N
      QHI(K,I,J)=QHI(K,I,J)+ALPHA(1)*W(K,1,I,L)*E2AH(L,J)+ALPHA(2)*W(K,2
     $,1,L)*EAH(I,J)
  274 CONTINUE
      QHI(K,I,J)=QHI(K,I,J)+W(K,3,I,J)
  273 CONTINUE
  272 CONTINUE
  283 CONTINUE
      DO 275 L=1,3
      DO 276 I=1,N
      DO 277 J=1,N
      DO 278 K=1,N
      PHI(L,I,J)=PHI(L,I,J)-AH3(I,K)*QHI(L,K,J)
  278 CONTINUE
  277 CONTINUE
  276 CONTINUE
  275 CONTINUE
  279 CONTINUE
      DO 280 K=1,3
      CALL GFN(G,H,N,Y,K,T,A)
      DO 281 I=1,N
      DO 282 J=1,N
      YN(I)=YN(I)+H*PHI(K,I,J)*G(J)
      IF(K .EQ. 3) YN(I)=YN(I)-ALPHA(1)*E2AH(I,J)*Y(1,J)-ALPHA(2)*EAH(I,
     $J)*Y(2,J)
  282 CONTINUE
  281 CONTINUE
  280 CONTINUE
      END
```

NONLINEAR MULTI-3-STEP

```
      SUBROUTINE STEP3(A,N,KSTEP,H,Y,UNIT,AH,AH2,AH3,AH4,EAH,E2AH,E3AH,P
     $HI,QHI,ALPHA,YN,INDEX,T,IS,HOLD)
      DIMENSION AH(35,35),AH2(35,35),AH3(35,35),AH4(35,35)
      DIMENSION UNIT(35,35),EAH(35,35),E2AH(35,35),E3AH(35,35)
      DIMENSION A(35,35),Y(4,35),PHI(4,35,35),QHI(4,35,35)
      DIMENSION W(4,4,35,35),G(35),ALPHA(1),YN(1),T(1)
      KUP=KSTEP
      IF(INDEX .EQ. 1) KUP=KSTEP+1
      DO 320 I=1,N
      YN(I)=0.0
  320 CONTINUE
      IF(H-HOLD) 341,340,341
  340 IF(IS .GT. 1) GO TO 321
  341 CONTINUE
      IF(N -1) 302,301,302
  301 EAH(1,1)=EXP(A(1,1)*H)
      E2AH(1,1)=EAH(1,1)*EAH(1,1)
      E3AH(1,1)=E2AH(1,1)*EAH(1,1)
      AH2(1,1)=AH(1,1)*AH(1,1)
      IF(INDEX .EQ. 1) GO TO 350
      AH3(1,1)=1./(AH2(1,1)*AH(1,1))
      GO TO 303
  302 CONTINUE
      DO 330 I=1,N
      DO 331 J=1,N
      DO 332 K=1,N
      AH2(I,J)=AH2(I,J)+AH(I,K)*AH(K,J)
  332 CONTINUE
  331 CONTINUE
  330 CONTINUE
      DO 333 I=1,N
      DO 334 J=1,N
      DO 335 K=1,N
      EAH(I,J)=EAH(I,J)+AH2(I,K)*AH(K,J)
  335 CONTINUE
  334 CONTINUE
  333 CONTINUE
      IF(INDEX .EQ. 1) GO TO 351
      CALL INVERT(EAH,N,AH3)
      CALL PADE(A,H,EAH,N)
      H2=H+H
      CALL PADE(A,H2,E2AH,N)
      H3=H2+H
      CALL PADE(A,H3,E3AH,N)
  303 CONTINUE
      DO 304 I=1,N
      DO 305 J=1,N
      W(1,1,I,J)=UNIT(I,J)-1.5*AH(I,J)+AH2(I,J)
      W(1,2,I,J)=UNIT(I,J)-0.5*AH(I,J)
      W(1,3,I,J)=UNIT(I,J)+0.5*AH(I,J)
      W(1,4,I,J)=UNIT(I,J)+1.5*AH(I,J)+AH2(I,J)
      W(2,1,I,J)=-2.*(UNIT(I,J)-AH(I,J))
      W(2,2,I,J)=-2.*UNIT(I,J)+AH2(I,J)
      W(2,3,I,J)=-2.*(UNIT(I,J)+AH(I,J))
```

```
      W(2,4,I,J)=-2.*UNIT(I,J)-4.*AH(I,J)-3.*AH2(I,J)
      W(3,1,I,J)=UNIT(I,J)-0.5*AH(I,J)
      W(3,2,I,J)=UNIT(I,J)+0.5*AH(I,J)
      W(3,3,I,J)=W(1,4,I,J)
      W(3,4,I,J)=UNIT(I,J)+2.5*AH(I,J)+3.*AH2(I,J)
  305 CONTINUE
  304 CONTINUE
  360 DO 306 II=1,KUP
      DO 307 I=1,N
      DO 308 J=1,N
      DO 309 K=1,N
      QHI(II,I,J)=QHI(II,I,J)+ALPHA(1)*W(II,1,I,K)*E3AH(K,J)+ALPHA(2)*W(
     $II,2,I,K)*E2AH(K,J)+ALPHA(3)*W(II,3,I,K)*EAH(K,J)
  309 CONTINUE
      QHI(II,I,J)=QHI(II,I,J)+W(II,4,I,J)
  308 CONTINUE
  307 CONTINUE
  306 CONTINUE
      DO 310 K=1,KUP
      DO 311 I=1,N
      DO 312 J=1,N
      DO 313 L=1,N
      IF(INDEX.EQ.0) PHI(K,I,J)=PHI(K,I,J)-AH3(I,L)*QHI(K,L,J)
      IF(INDEX.EQ.1) PHI(K,I,J)=PHI(K,I,J)-AH4(I,L)*QHI(K,L,J)
  313 CONTINUE
  312 CONTINUE
  311 CONTINUE
  310 CONTINUE
  321 CONTINUE
      DO 314 K=1,KUP
      CALL GFN(G,H,N,Y,K,T,A)
      DO 315 I=1,N
      DO 316 J=1,N
      YN(I)=YN(I)+H*PHI(K,I,J)*G(J)
      IF(K .EQ. KUP) YN(I)=YN(I)-ALPHA(3)*EAH(I,J)*Y(3,J)-ALPHA(2)*E2AH(
     $I,J)*Y(2,J)-ALPHA(1)*E3AH(I,J)*Y(1,J)
  316 CONTINUE
  315 CONTINUE
  314 CONTINUE
      RETURN
  350 AH3(1,1)=AH2(1,1)*AH(1,1)
      AH4(1,1)=1.0/(AH3(1,1)*AH(1,1))
      GO TO 357
  351 DO 352 I=1,N
      DO 353 J=1,N
      AH3(I,J)=EAH(I,J)
      EAH(I,J)=0.0
  353 CONTINUE
  352 CONTINUE
      DO 354 I=1,N
      DO 355 J=1,N
      DO 356 K=1,N
      EAH(I,J)=EAH(I,J)+AH3(I,K)*AH(K,J)
  356 CONTINUE
  355 CONTINUE
  354 CONTINUE
      CALL INVERT(EAH,N,AH4)
      CALL PADE(A,H,EAH,N)
```

```
      H2=H+H
      CALL PADE(A,H2,E2AH,N)
      H3=H2+H
      CALL PADE(A,H3,E3AH,N)
  357 CONTINUE
      DO 358 I=1,N
      DO 359 J=1,N
      W(1,1,I,J)=-0.1*UNIT(I,J)+1.1*AH(I,J)-23.*AH2(I,J)/15.+AH3(I,J)
      W(1,2,I,J)=-0.1*UNIT(I,J)+AH(I,J)-29.*AH2(I,J)/60.
      W(1,3,I,J)=-0.1*UNIT(I,J)+0.9*AH(I,J)+7.*AH2(I,J)/15.
      W(1,4,I,J)=-0.1*UNIT(I,J)+0.8*AH(I,J)+79.*AH2(I,J)/60.+.9*AH3(I,J)
      W(2,1,I,J)=0.3*UNIT(I,J)-2.3*AH(I,J)+2.1*AH2(I,J)
      W(2,2,I,J)=0.3*UNIT(I,J)-2.*AH(I,J)-0.05*AH2(I,J)+AH3(I,J)
      W(2,3,I,J)=0.3*UNIT(I,J)-1.7*AH(I,J)-1.9*AH2(I,J)
      W(2,4,I,J)=0.3*UNIT(I,J)-1.4*AH(I,J)-3.45*AH2(I,J)-2.7*AH3(I,J)
      W(3,1,I,J)=-0.3*UNIT(I,J)+1.3*AH(I,J)-0.6*AH2(I,J)
      W(3,2,I,J)=-0.3*UNIT(I,J)+AH(I,J)+.55*AH2(I,J)
      W(3,3,I,J)=-0.3*UNIT(I,J)+0.7*AH(I,J)+1.4*AH2(I,J)+AH3(I,J)
      W(3,4,I,J)=-0.3*UNIT(I,J)+0.4*AH(I,J)+1.95*AH2(I,J)+2.7*AH3(I,J)
      W(4,1,I,J)=0.1*(UNIT(I,J)-AH(I,J))+AH2(I,J)/30.
      W(4,2,I,J)=0.1*UNIT(I,J)-AH2(I,J)/60.
      W(4,3,I,J)=0.1*(UNIT(I,J)+AH(I,J))+AH2(I,J)/30.
      W(4,4,I,J)=0.1*UNIT(I,J)+0.2*AH(I,J)+11.*AH2(I,J)/60.+0.1*AH3(I,J)
  359 CONTINUE
  358 CONTINUE
      GO TO 360
      END
```

LISTS OF SYMBOLS AND DEFINITIONS

Indices

$i, j, K, \ell, m, n, N, p, q, \nu$

Scalars

$G, E, k, K^{\#}, L, L^*, L^{\#}, Q^{\#}, \mathfrak{z}$

$\alpha, \hat{\alpha}, \beta, \gamma, \delta(h), \epsilon, \zeta, \hat{\zeta}, \theta, \kappa, \xi, \sigma, \phi, \Gamma, \Lambda$

Vectors

$\mathbf{e}, \mathbf{f}, \mathbf{g}, \mathbf{G}, \mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{y}', \mathbf{y}^*, \mathbf{z}, \boldsymbol{\phi}, \Psi, \theta^g$

Matrices

$\mathbf{A}, \mathbf{C}(h), \mathbf{E}, \mathbf{H}, \mathbf{K}, \mathbf{Q}, \xi, \boldsymbol{\phi}_{Ki}(\mathbf{A}h)$

Symbols

!	factorial
$\in$	belongs to
$\lvert\ \rvert$	absolute value
$\lVert\ \rVert$	norm
>>	much greater than
$\exists$	there exists
$\ni$	such that
$\forall$	for every
$\prod$	product
$\sum$	summation
LHS, RHS	left-hand side, right-hand side

LMS	Linear Multistep
NLMS	Nonlinear Multistep
t	time variable
t'	integration variable
h	step size
h_N	step size of method N
ζ	roots of characteristic polynomial
p_N	method N of order p
$\mathbf{y}_o$	initial vector
$\mathbf{y}_n, \mathbf{g}_n, \mathbf{f}_n$	values of $\mathbf{y}, \mathbf{g}, \mathbf{f}$ evaluated at t_n
$\mathbf{g}^{(j)}$	j-th derivative of $\mathbf{g}$
$\rho^{\#}(\tau)$	a polynomial in τ
$\mathcal{P}_p$	class of polynomials of degree p
$G(\mathbf{y})$	a function of $\mathbf{y}$
$\mathbf{I}_i^j(\mathbf{A}h)$	an integral
$\mathcal{J}_i^j$	a function of $\mathbf{I}_i^j(\mathbf{A}h)$
$\omega(\epsilon)$	M.O.C. (modulus of continuity)
$\lambda(\mathbf{A})$	eigenvalues of $\mathbf{A}$
$\rho(\mathbf{A})$	spectral radius of $\mathbf{A}$
$\rho(\zeta), \rho(\xi), \rho(\lambda,\zeta)$	characteristic polynomials
$O(h^{p+2})$	order of h^{p+2}
C^{p+1}	functions which have (p+1)-th continuous derivative
$\mathbf{C}_{p+1}$	error function

$\mathcal{L}[y(t)]$	true operator
$\mathcal{L}_N[y(t);h]$	nonlinear operator
$\tau[y(t);h]$	local discretization error

REFERENCES

1. Bachman, G. and L. Narici (1966), Functional Analysis, Academic Press, New York.

2. Bailey, H. E. (1969), "Numerical Integration of the Equations Governing the One-Dimensional Flow of a Chemically Reactive Gas," The Physics of Fluids, 12, No. 11, pp. 2292-2300.

3. Bjurel, G. (1971), "Modified Multistep Methods for a Class of Stiff Ordinary Differential Equations," Dept. of Information Processing, The Royal Institute of Technology, Stockholm, Report No. NA 71.42.

4. Bjurel, G. (1971), "Supplement to Report on Modified Linear Multistep for a Class of Stiff Ordinary Differential Equations," Dept. of Information Processing, The Royal Institute of Technology, Stockholm, Report No. 71.43.

5. Blue, J. L. and H. K. Gummel (1970), "Rational Approximations to Matrix Exponential for Systems of Stiff Equations," Jour. of Computational Physics, 5, pp. 70-83.

6. Calahan, D. A. (1967), "Numerical Solution of Linear Systems with Widely Separated Time Constants," Proceedings of IEEE, pp. 2016-2017.

7. Certaine, J. (1960), "The Solution of Ordinary Differential Equations with Large Time Constants," in Math. Methods for Digital Computers, ed. A. Ralston and H. Wilf, John Wiley and Sons, Inc., New York, pp. 128-132.

8. Cohen, E. R. (1958), "Some Topics in Reactor Kinetics," Proc. 2nd UN Int. Conf. PAUSE, 11, pp. 302-309.

9. Curtiss, C. F. and J. O. Hirschfelder (1952), "Integration of Stiff Equations," Proc. of the Nat. Acad. of Sc., 38, pp. 235-243.

10. Dahlquist, G. (1963), "A Special Stability Problem for Linear Multistep Methods," BIT, 3, pp. 27-43.

11. Ehle, B. L. (1972), "A Comparison of Numerical Methods for Solving Certain Stiff Ordinary Differential Equations," Dept. of Math., University of Victoria, Canada, Report No. 70.

12. Forsythe, G. and C. B. Moler (1967), Computer Solution of Linear Algebraic Systems, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

13. Fowler, M. E. and R. M. Warten (1967), "Numerical Integration Technique for Ordinary Differential Equations with Widely Separated Eigenvalues," IBM Journal, 11, No. 5, pp. 537-543.

14. Gear, C. W. (1971), Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

15. Guderley, K. G. and C. C. Hsu(1972), "A Predictor-Corrector Method for a Certain Class of Stiff Differential Equations," Math. of Computation, 26, No. 117, pp. 51-69.

16. Henrici, P. (1962), Discrete Variable Methods in Ordinary Differential Equations, John Wiley and Sons, Inc., New York.

17. Henrici, P. (1963), Error Propagation for Difference Methods, John Wiley and Sons, Inc., New York.

18. Hochstadt, H. (1964), Differential Equations, Holt, Rinehart and Winston, New York.

19. Hull, T. E., W. H. Enright, B. M. Fellen, and A. E. Sedgwick (1972), "Comparing Numerical Methods for Ordinary Differential Equations," SINUM, 9, No. 4, pp. 603-637.

20. Isaacson, E. and H. B. Keller (1966), Analysis of Numerical Methods, John Wiley and Sons, Inc., New York.

21. Jain, R. K. (1972), "Some A-Stable Methods for Stiff Ordinary Differential Equations," Math. of Computation, 26, No. 117, pp. 71-77.

22. H. B. Keller (1968), Numerical Methods for Two-Point Boundary-Value Problems, Blaisdell Pub. Co., Waltham, Massachusetts.

23. Lambert, J. D. and S. T. Sigurdsson (1972), "Multistep Methods with Variable Matrix Coefficients," SINUM, 9, No. 4, pp. 715-733.

24. Lawson, J. D.(1967), "Generalized Runge-Kutta Processes for Stable Systems with Large Lipschitz Constants," SINUM, 4, pp. 372-380.

25. Lawson, J. D. (1967), "An Order Six Runge-Kutta Process with Extended Region of Stability," SINUM, 4, No. 4, pp. 620-625.

26. Little, W. W., Jr., K. F. Hansen, E. A. Mason, and B. V. Koen (1964), "A Stable Numerical Solution of the Reactor Kinetics Equations," Trans. Amer. Nuclear Society, 7, No. 1, pp. 3-4.

27. Miranker, W. L. (1971), "Matrical Difference Schemes for Integrating Stiff Systems of Ordinary Differential Equations," Math. of Computation, 25, No. 116, pp. 717-728.

28. Norsett, S. P. (1969), "An A-Stable Modification of the Adams-Bashforth Methods," Conference on the Numerical Solution of D. Equations, 109, Springer-Verlag.

29. Seinfeld, J. H., L. Lapidus, and M. Hwang (1970), "Review of Numerical Integration Techniques for Stiff Ordinary Differential Equations," Ind. Eng. Chem. Fundam., 9, No. 2, pp. 266-275.

30. Treanor, C. E. (1966), "A Method for the Numerical Integration of Coupled First-Order Differential Equations with Greatly Different Time Constants," Math. of Computation, 20, No. 93, pp. 39-45.

31. Varga, R. S. (1962), Matrix Iterative Analysis, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

32. Widlund, O. B. (1967), "A Note on Unconditionally Stable Linear Multistep Methods," BIT, 7, pp. 65-70.

33. Young, D. M. (1971), Iterative Solution of Large Linear Systems, Academic Press, New York.